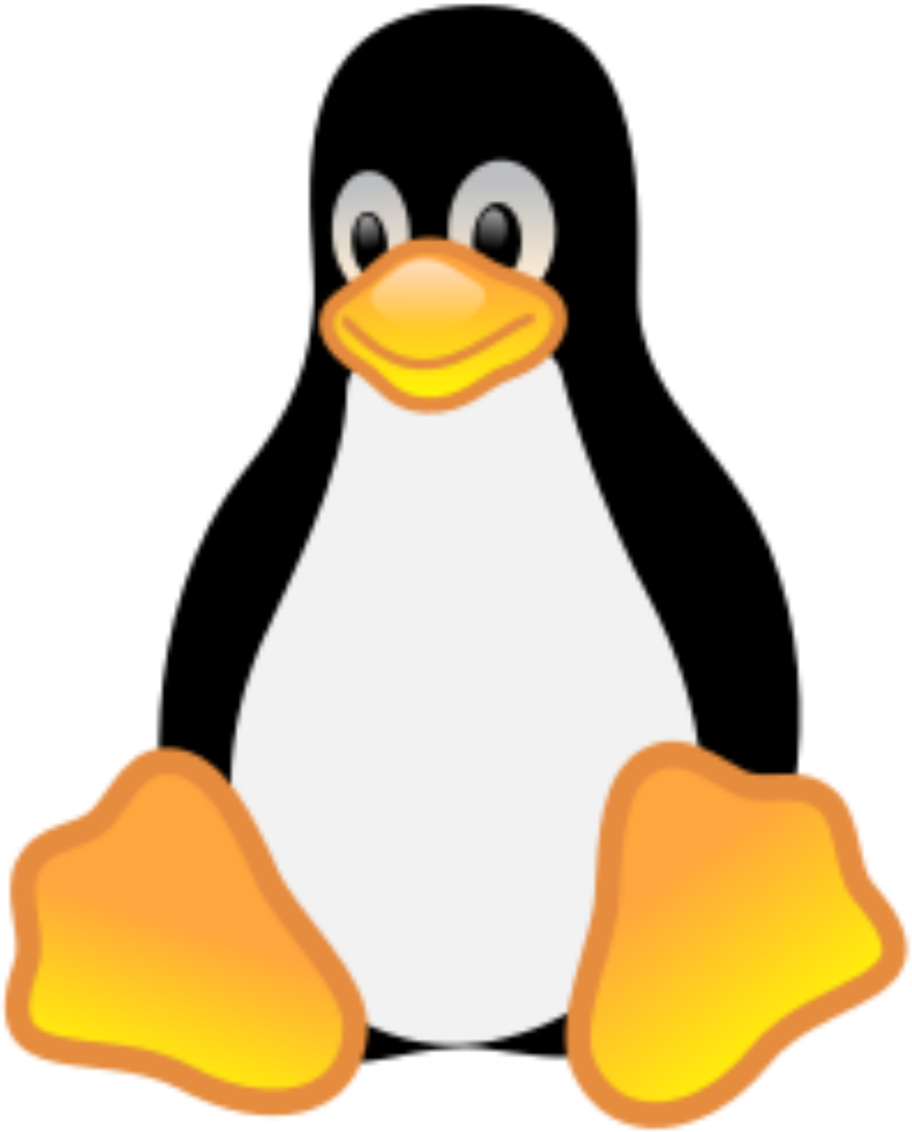
Kendi Linux'unu Yap

Masaüstü Ortamı

LXDE



Yazar

- Bayram KARAHAN
- Celalettin AKARSU

Paket Derleme

- Bayram KARAHAN
- Celalettin AKARSU

İletişim

- https://github.com/kendilinuxunuyap
- https://kendilinuxunuyap.github.io/wm
- kendilinuxunuyap@gmail.com

ISBN: 978-625-90289-0-3

Yayın Yılı: Nisan 2026

Ön Söz

Bu kitap, açık kaynak ve özgür yazılım dünyasına ilgi duyan herkes için hazırlanmıştır. Amacı, Türkçe kaynak eksikliğini gidermek ve kendi Linux dağıtımını oluşturmak isteyenlere yol göstermektir.

Bu serinin ilk kitabında, **Temel Linux Sistemi**'nin nasıl derlenip kullanılabilir hale getirileceği anlatılmıştı.

İkinci kitabında, oluşturduğumuz **Temel Linux Sistemi** üzerinde **X Pencere Sistemi** nasıl derlenip çalıştırılacağı anlatıldı.

Bu kitabımızda ise **X Pencere Sistemi** üzerine **LXDE** masaüstü ortamının derlenmesi ve yüklenmesi anlatılmaktadır. **LXDE** masaüstü ortamının açılabilmesi için **Display Manager (Giriş Ekranı + Oturum Başlatıcı)** derlenerek sistemin açılışı kullanıcı dostu hale getirilecektir.

Bu kitap, LXDE masaüstü ortamının kaynaktan derlemek isteyen Linux kullanıcıları için kapsamlı bir rehber sunmayı amaçlamaktadır. Ayrıca, bu süreç Linux sistemlerinin nasıl çalıştığını anlamak için rehber olacaktır.

İçindekiler

1- Temel Kavramlar	5
2- X Pencere Sistemin Hazırlanması	6
3- GNU Araçlarıyla Pencere Yöneticisi Derleme	16
4- ISO Hazırlama	101
5- Oluşan Sistemin Çalıştırılması ve İncelenmesi	104
6- Lxde Masaüstü Ortamı Çalıştırma	109
7- Yardımcı Konular	111
8- Kaynaklar	163
9- Geliştiricilere Mesajımız	164

Temel Kavramlar

Window Manager (Pencere Yöneticisi)

Xorg yalnızca pencereyi oluşturur. Pencereyi taşıma, kapatma, büyütme ve küçültme işlemlerini yapar. Basit ortam sunan pencere yöneticileri(Openbox, i3).

Xorg + Openbox => Grafik ortamda çalışan bir sistem oluşturur.

Openbox kendi başına bir masaüstü ortamı değil ancak başka bileşenlerle basit bir masaüstü ortamı gibi çalışabilir.

Display Manager (Giriş Ekranı + Oturum Başlatıcı)

Sistem açıldığında kullanıcı seçimi ve parola ekranı sunar. Girişten sonra kullanıcı oturumunu (masaüstü yöneticisini) başlatır.

Tercih edilen giriş ekranı yöneticileri(gdm, lightdm, lxdm). X Pencere Sistemi için zorunlu değildir.

Desktop Manager (Masaüstü Yöneticisi)

Masaüstü ortamını sağlayan uygulamalar bütünüdür. Dosya yöneticisi, pencere yöneticisi, ayarlar ve panel gibi bileşenleri içerir.

Tercih edilen masaüstü yöneticisi;

1. **LXDE (Lightweight X11 Desktop Environment)**

   LXDE, GTK tabanlı az kaynak kullanan bir masaüstü ortamıdır. Pencere yöneticisi Openbox dır.

2. **XFCE**

   XFCE, GTK tabanlı hafif ve kullanımı kolay bir masaüstü ortamıdır.

3. **KDE Plasma**

   KDE, Qt tabanlı tabanlı gelişmiş bir masaüstü ortamıdır.

4. **GNOME**

   GNOME, modern bir masaüstü ortamıdır. Dokunmatik ekranlı cihazları desteklenmektedir.

5. **Cinnamon**

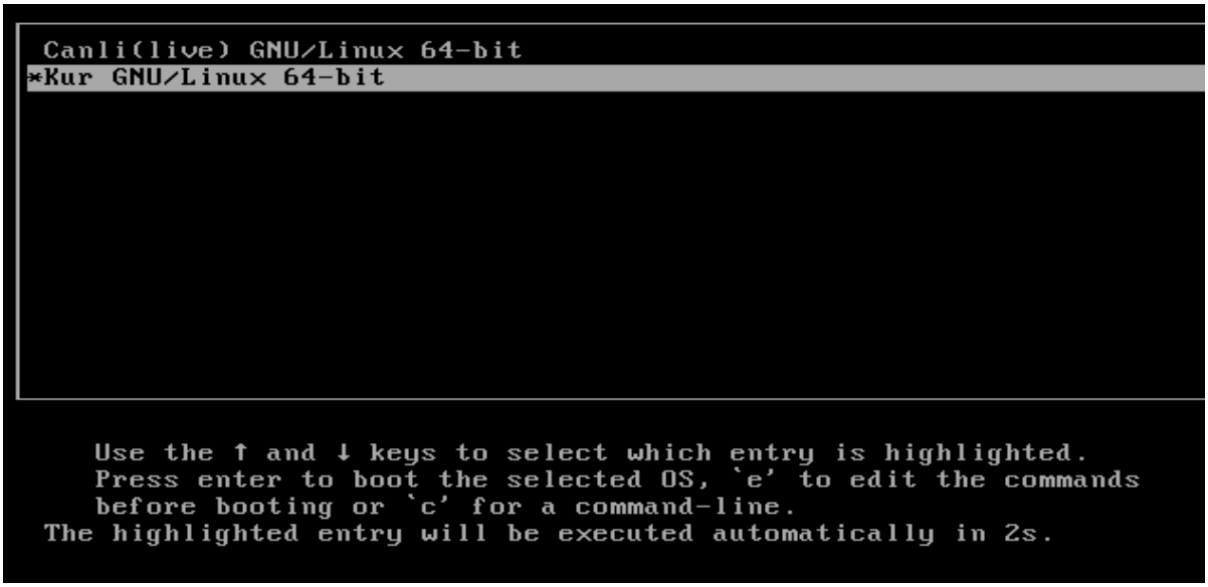
   Cinnamon, GTK tabanlı ek bileşenleri python ile yapılmış modern masaüstü ortamdır. Linux Mint tarafından geliştirilmektedir. GNOME 3'ün bir çatallanmasıdır.

X Pencere Sistemin Hazırlanması

Bir önceki kitabımızda **X Pencere Sistemini** oluşturmayı ve iso halinde kurulumu anlatmıştık. Şimdi ise bu **X Pencere Sistemini** kurarak bu sistem üzerine **LXDE** araçları ve kütüphanelerini derleyerek **LXDE** masaüstü ortamının nasıl çalışacağı anlatılacaktır. **X Pencere Sistemi** https://github.com/kendilinuxunuyap/kly-x11-distro/releases/download/current/kly-x11-distro.iso adresinde bulunmaktadır. İso indirip kurulum yapabilirsiniz.

X Pencere Sistemi Kurulumu

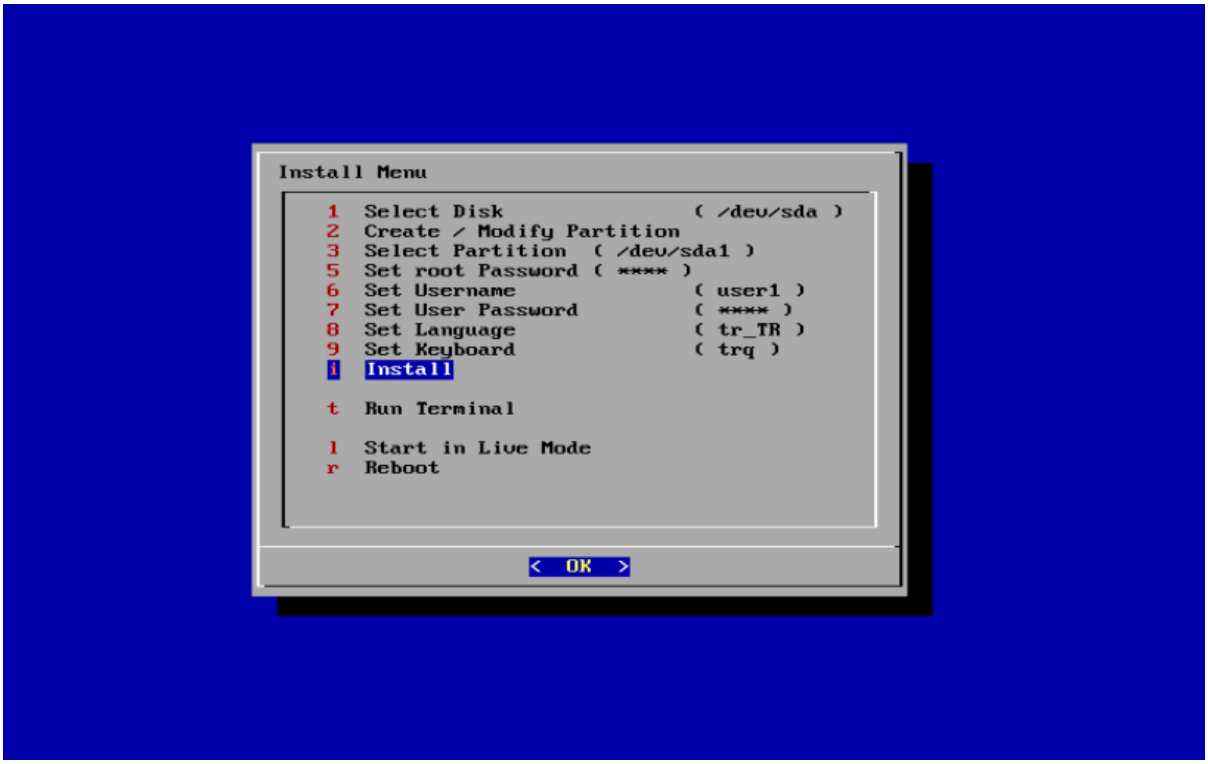
Sistemin kurulumu için resimlerde görünen sıraya göre seçimler yapmalıyız.



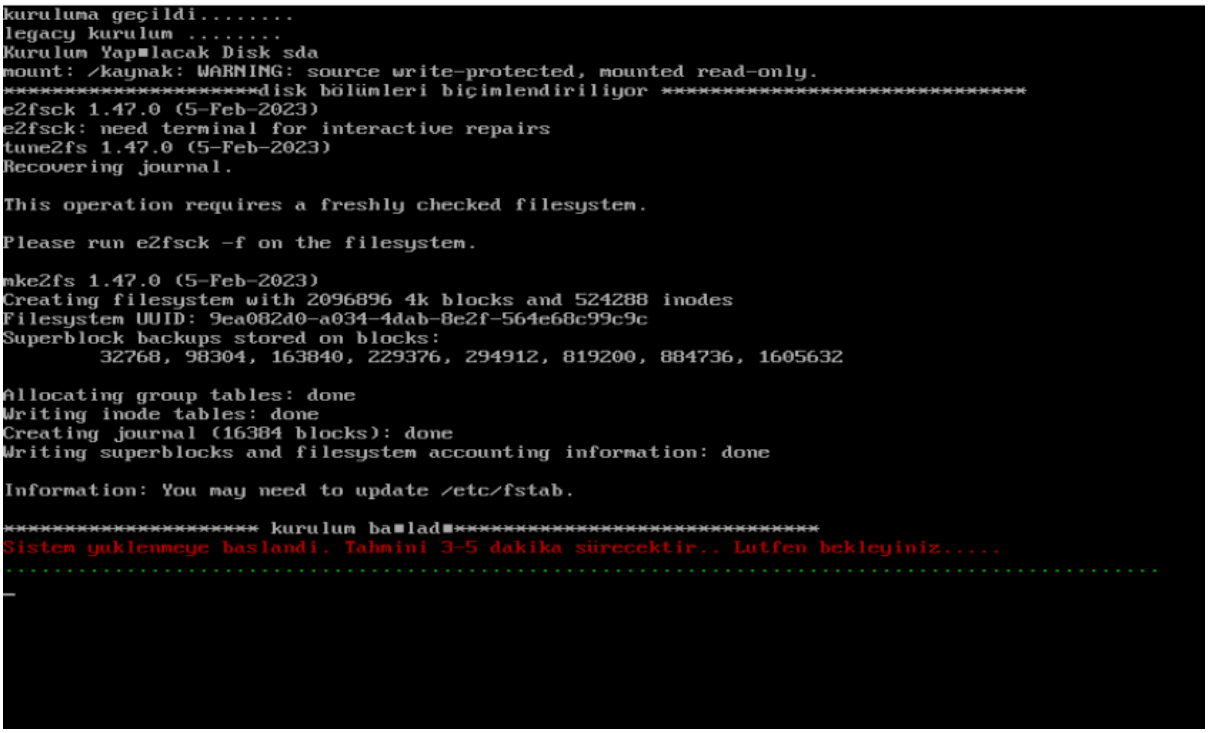
Kurulum menüsünde kullanıcı adları ve parolaları, klayve varsayılan olarak;

- Kullanıcı: root Parola: 1
- Kullanıcı: user1 Parola: 1
- Dil : tr_TR
- Klavye : trq

menüden değişiklik yapabilirsiniz. Değişiklik yapmadan sadece kurulum diskini ve disk bölümünü seçip Install(Yükle) işlemi yapabilirsiniz.

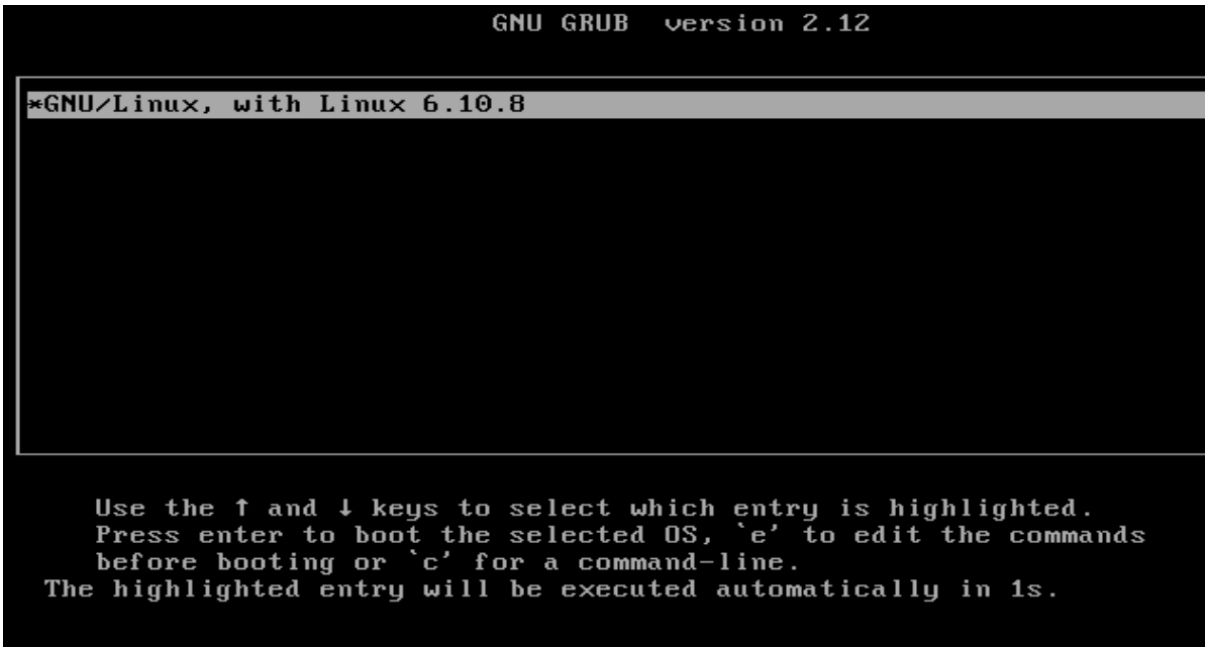


kly [Çalışıyor] - Oracle VM VirtualBox

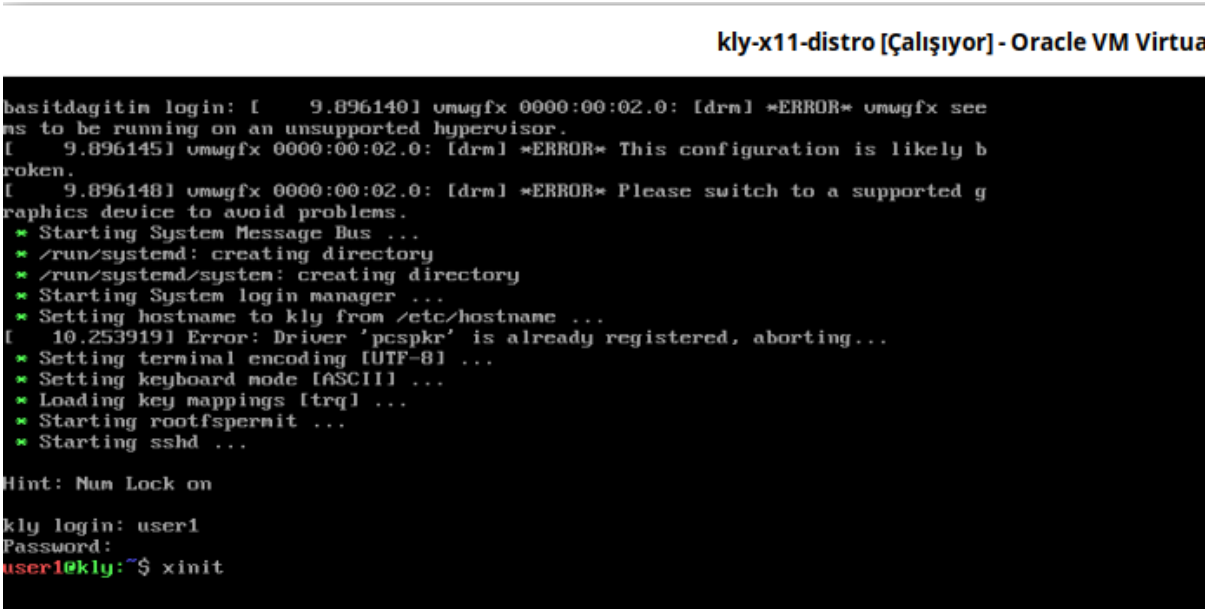


Sistemin Çalışması

Sistem kurulumu gerçekleştiğinde sistem resimde görüldüğü gibi açılmalıdır.

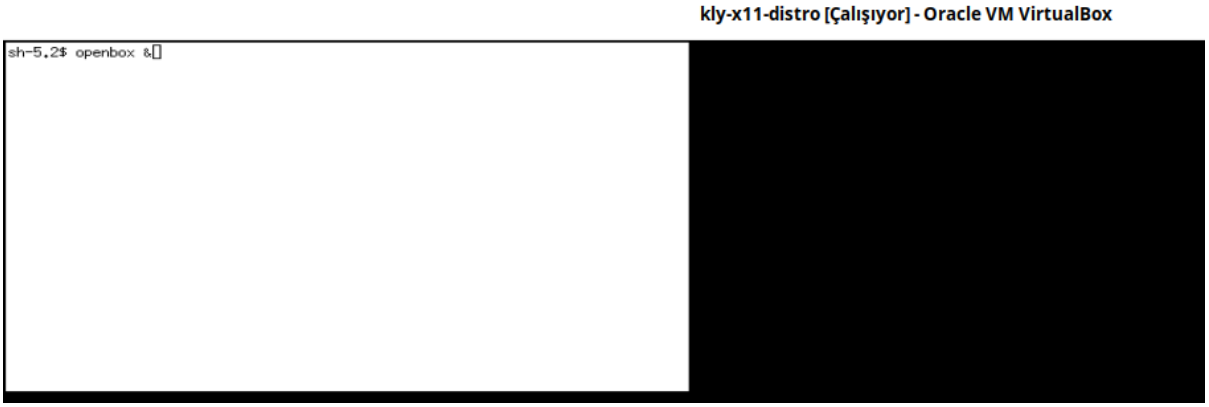


Sisteme **user1** (parola=1) kullanıcısı olarak giriş yapıldığı görülmektedir.

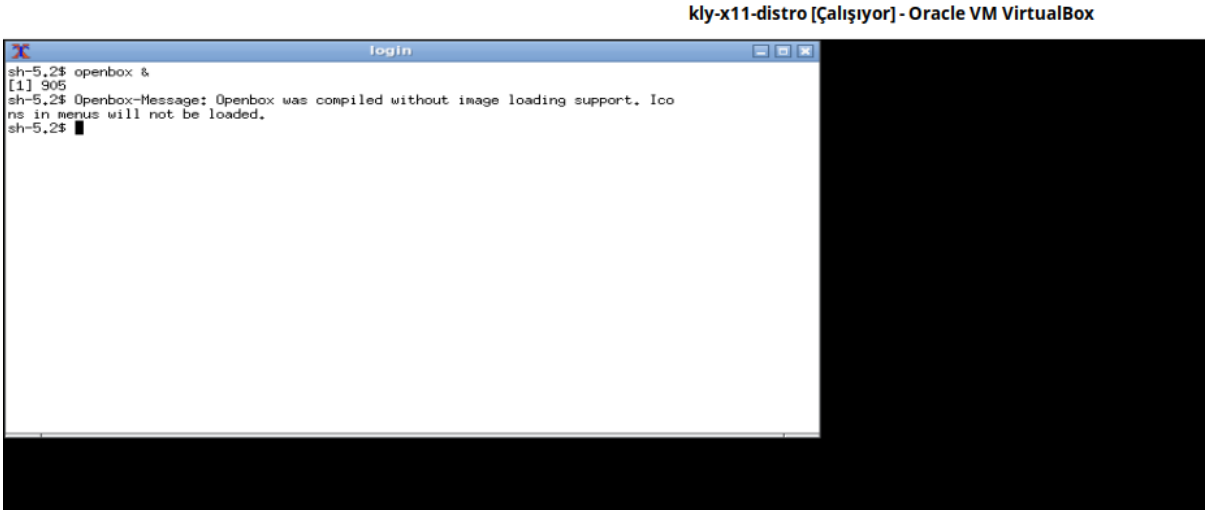


xinit çalışınca **X Pencere Sistemine** geçiş yapılacak.

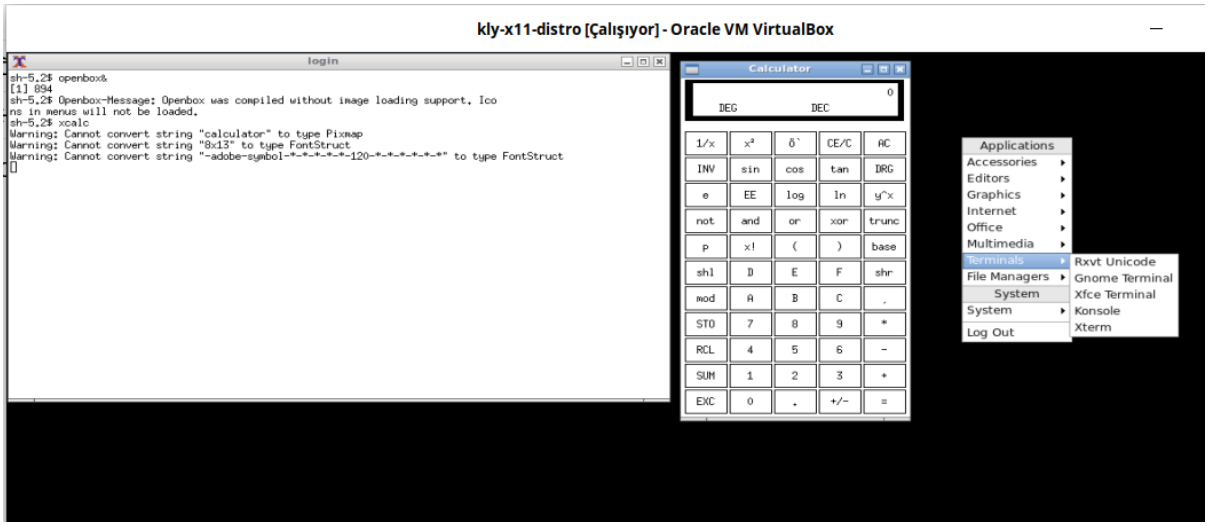
xinit çalışınca **X Pencere Sistemi** aşağıdaki gibi açılacaktır.



X Pencere Sistemi çalıştırılmış ve pencere yönetimini ve masaüstü ortamı için **openbox** çalıştırılıyor.



xcalc uygulamasının çalıştırılması aşağıda gösterilmiştir.

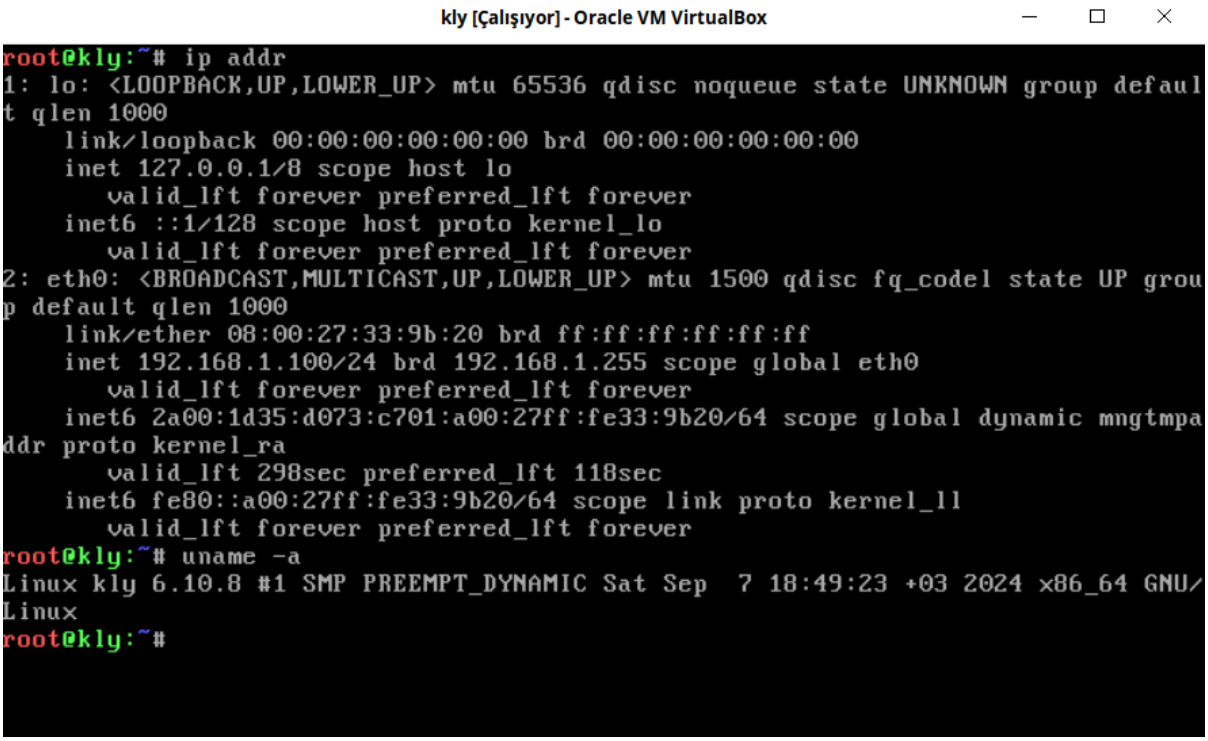


X Pencere Sistemi sistemi görüldüğü üzere çalışmaktadır. Artık **LXDE Masaüstü Ortamı** paketlerini ve bağımlılıklarını **kly Paket Sistemini** kullanarak derlenecek ve **X Pencere Sistemi** üzerine kurularak masaüstü ortamımızı çalışır hale getireceğiz.

Sistemin Internet Bağlantısı

Temel Sistem içerisinde bulunan **iproute2, dhclient, net-tools** paketleri derlendi ve açılışta **dhclient** aktif hale gelmektedir. Bu paket ile ağa dahil olunmaktadır.

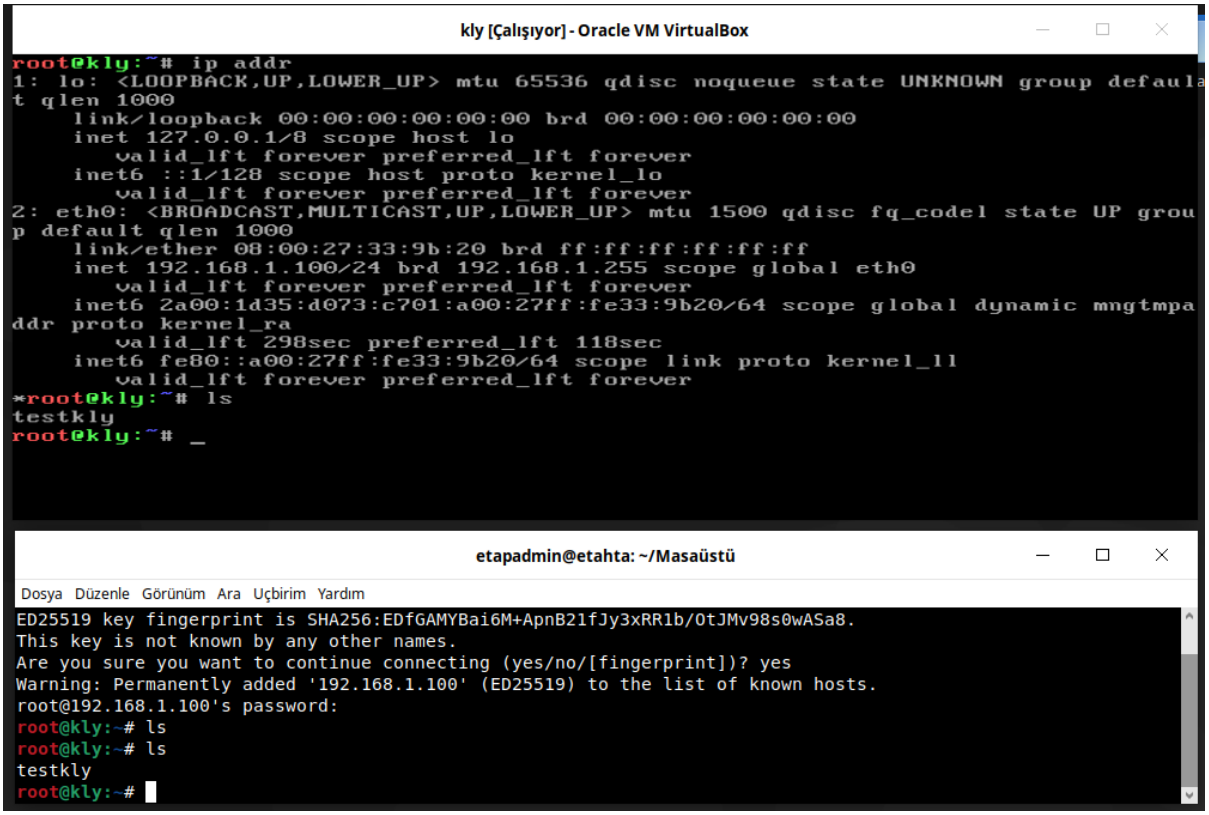
Aşağıda atanan ip adresini görmekteyiz. Eğer ip adresi almamışsa terminalde **dhclient** komutunu çalıştırınız.



Sisteme openssh ile Bağlanma

ssh terminal üzerinden uzak bilgisayarlara erişim yapan bir uygulamadır. **Temel Sistem** içerisinde ssh paketi derlendi ve açılışta aktif hale gelmektedir.

Aşağıda **ssh** ile erişimin nasıl yapıldığı görülmektedir.

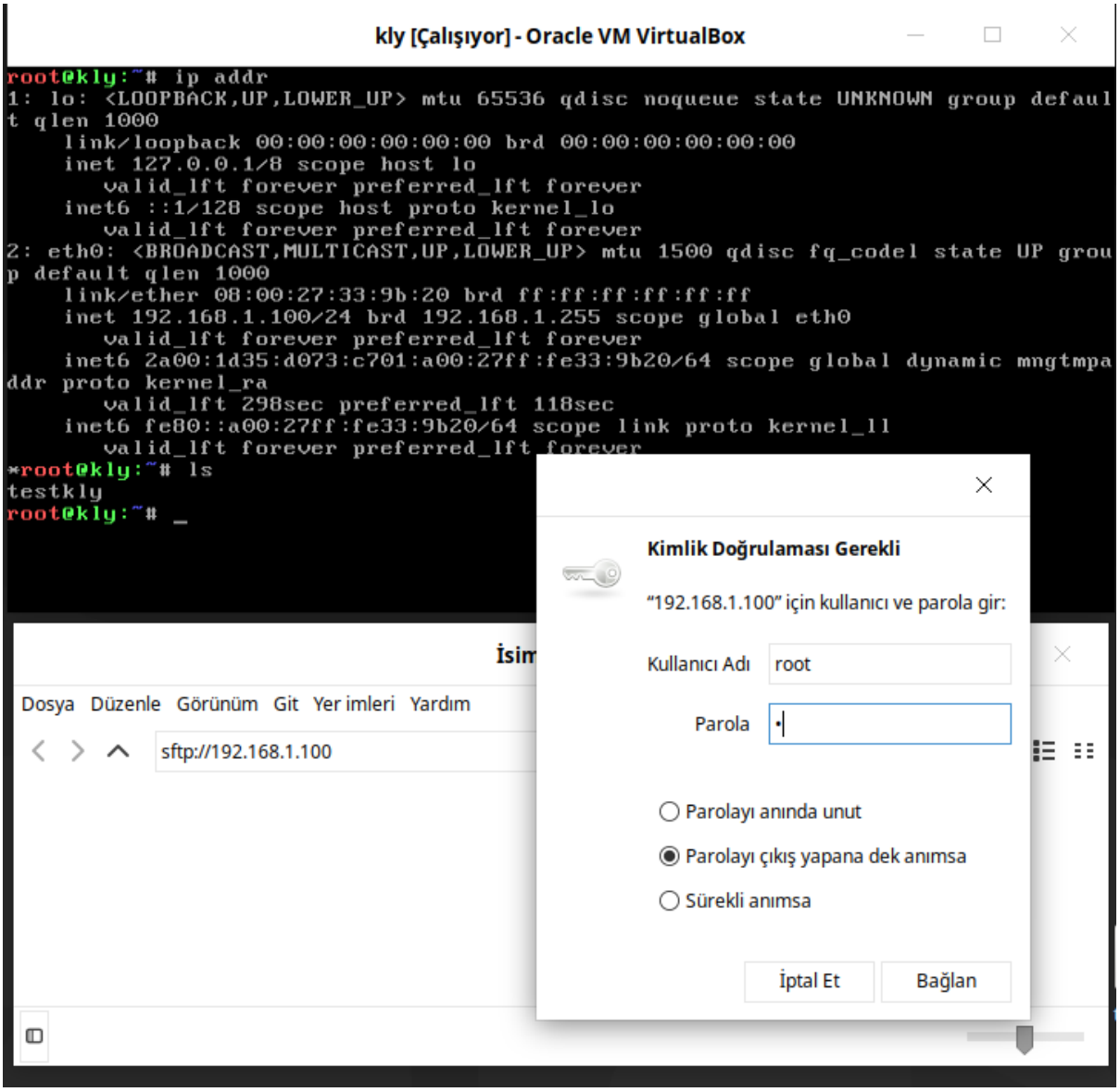


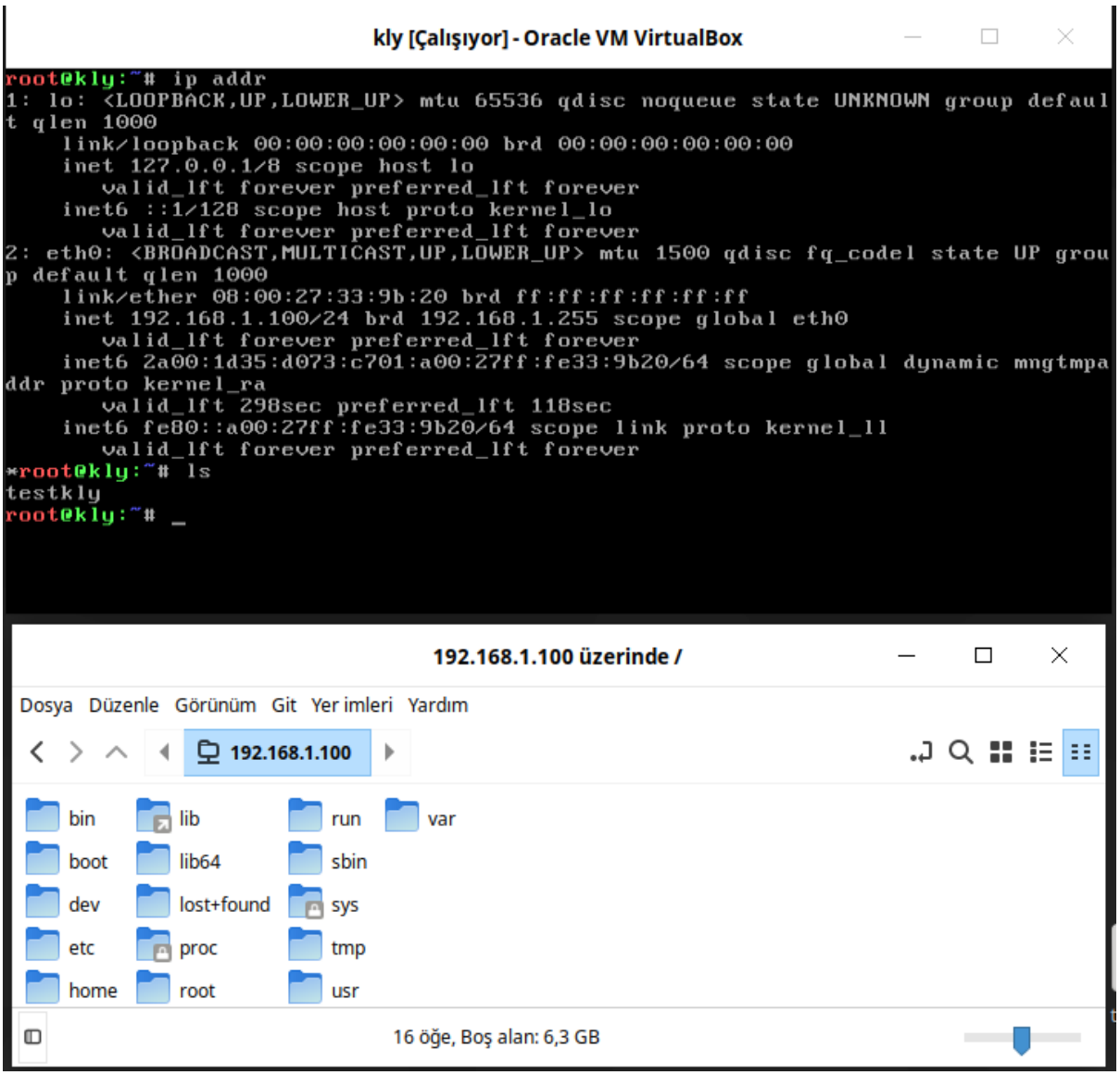
Bu doküman boyunca paketleri **Debian** ortamında derleyeceğiz. Derlediğimiz paketleri **X Pencere Sistemi'ne** kopyalacağız. Kopyalama işlemini **ssh** paketi içinde gelen **sftp** veya **scp** ile yapacağız. Kopyalanan paketlerin **kurulması ve kaldırıması** gibi işlemleri **ssh** bağlantısı üzerinden gerçekleştireceğiz.

Sisteme sftp ile Bağlanma

sftp terminal veya pencere yöneticisi üzerinden uzak bilgisayarlara erişim yapan bir uygulamadır. **Temel Sistem** içerisinde **ssh** paketinin bir parçası olarak gelmektedir.

Aşağıda **sftp** ile erişimin nasıl yapıldığı görülmektedir.

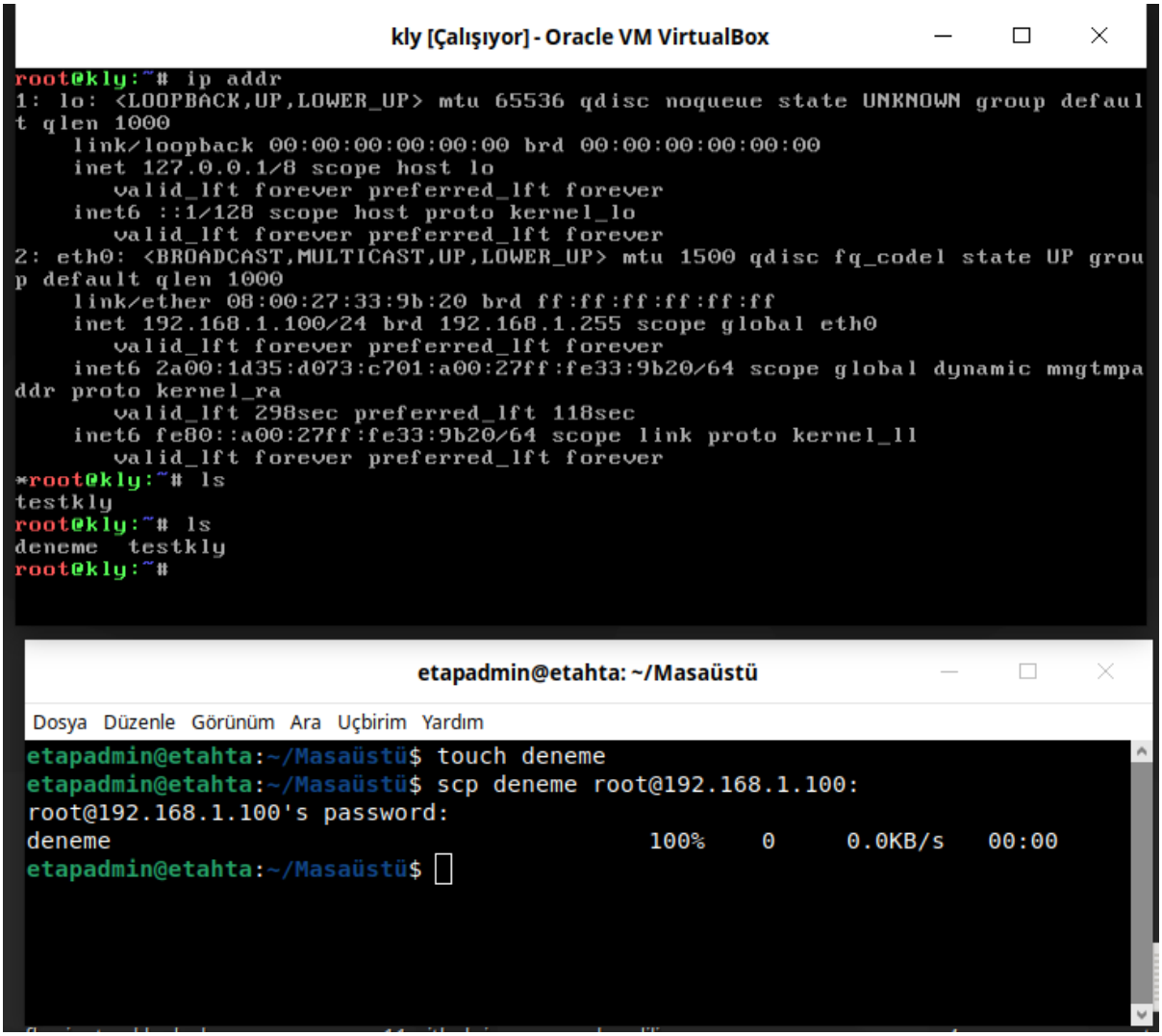



kly [Çalışıyor] - Oracle VM VirtualBox
root@kly:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host proto kernel_lo
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
link/ether 08:00:27:33:9b:20 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.100/24 brd 192.168.1.255 scope global eth0
valid_lft forever preferred_lft forever
inet6 2a00:1d35:d073:c701:a00:27ff:fe33:9b20/64 scope global dynamic mngtmpa
ddr proto kernel_ra
valid_lft 298sec preferred_lft 118sec
inet6 fe80::a00:27ff:fe33:9b20/64 scope link proto kernel_ll
valid_lft forever preferred_lft forever
*root@kly:~# ls
testkly
root@kly:~# _
192.168.1.100 üzerinde /
Dosya Düzenle Görünüm Git Yer imleri Yardım
192.168.1.100
bin
lib
run
var
boot
lib64
sbin
dev
lost+found
sys
etc
proc
tmp
home
root
usr
16 öğe, Boş alan: 6,3 GB

Sisteme scp ile Dosya Kopyalama

scp terminal üzerinden uzak bilgisayarlara dosya kopyalaması yapan bir uygulamadır. **Temel Sistem** içerisinde **ssh** paketinin bir parçası olarak gelmektedir.

Aşağıda **scp** ile **deneme** dosyasının nasıl kopyalandığı görülmektedir.



GNU Araçlarıyla Pencere Yöneticisi Derleme

Ön Hazırlık

Paket derleme işlemi öncesi aşağıdaki konuları bilmemiz gerekmektedir. Bunlar;

1. Derleme(Dinamik/Static)
2. chroot Kullanımı
3. İso Oluşturma
4. ssh Kullanımı
5. sftp Kullanımı
6. scp Kullanımı
7. VirtualBox Kullanmı
8. cfdisk Kullanımı

Burada liste halinde verilen konu başlıkları bu dokümanın **Yardımcı Konular** bölümünde anlatılmaktadır.

Bundan sonraki adımlarda kendi dağıtımımızın **LXDE masaüstü ortmaını** derleyerek **X Pencere Sistemi** üzerinde çalıştıracağız!

GNU Araçlarıyla **LXDE Masaüstü Ortamının** derleme işlemini **kly Paket Sistemi** kullanılarak derleyeceğiz. Derleme işlemini **kly -c** komutuyla yapacağız. Derlenen paketleri **scp** ve **sftp** kullanarak **Temel Sistem** üzerine kopyalayacağız. **kly -pi** komutumuzla kopyaladığımız paketi **X Pencere Sistemi** üzerine kuracağız. Oluşturduğumuz paketleri istersek github'a yükleyip. github üzerinden kururabiliriz.

LXDE Masaüstü Ortamı için Gerekli Paketler

0- Ön Hazırlık	29- libXfont	58- cups
1- libfm	30- libxkbcommon	59- gdbm
2- menu-cache	31- libxkbui	60- mpdecimal
3- libfm-extra	32- libxklavier	61- libgusb
4- lxmenu-data	33- libXpresent	62- libisl
5- lxde-common	34- libXres	63- libmpc
6- lxappearance	35- libxss	64- duktape
7- lxinput	36- libXtst	65- atkmm
8- lxrandr	37- libXv	66- at-spi2-core
9- lxsession	38- libXvMC	67- libtiff
10- lxpanel	39- libXxf86dga	68- libepoxy
11- lxtask	40- libXxf86vm	69- gobject-introspection
12- lxterminal	41- tslib	70- p11-kit
13- pcmanfm	42- vte3	71- nettle
14- lxlauncher	43- xapp	72- desktop-file-utils
15- lxhotkey	44- xcb-util-cursor	73- libidn2
16- lxde-icon-theme	45- xcb-util-errors	74- locale-tr
17- libjpeg-turbo	46- xcb-util-image	75- hicolor-icon-theme
18- cairo	47- xcb-util-keysyms	76- polkit
19- gdk-pixbuf	48- xcb-util-renderutil	77- libjpeg62
20- gtksourceview4	49- xcb-util-wm	78- gpicview
21- hsakmt	50- xtrans	79- libdmx
22- libFS	51- libkeybinder3	80- lightdm
23- libnotify	52- libexif	81- lightdm-gtk-greeter
24- libvdpau	53- gtk3	82- libgpg-error
25- libwnck3	54- shared-mime-info	83- libgcrypt
26- libXaw3d	55- lz4	84-
27- libXcomposite	56- gnutls	85-
28- libXdamage	57- lcms2	86-

Bağımlılık Zinciri

LXDE paketleri **gtk3** ile derlenecek. Bunun için ilk olarak **libfm** paketinin **gtk3** ile derlenmeli ve sonra mevcut sistemimize kurmalıyız. Sistemimize kurulduktan sonra **LXDE** paketlerini **gtk3** ile derlenmeli. **libfm** ve **menu-cache** ilk kurulması gereken paketler. Paket derleme işlemine başlamadan önce, aşağıdaki temel araçları sisteminize kurmalısınız.

```
sudo apt update
sudo apt-get install debootstrap xorriso mtools make squashfs-tools gcc wget unzip xz-utils tar zstd fakeroot \
autoconf automake autotools-dev make meson cmake ninja-build pkgconf patch libtool grub-pc grub-pc-bin
```

libfm

Lxde için menü tanımları ve veri dosyalarını içerir. İlk derlenemesi greken pakettir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libfm"
version="1.4.0"
description="Library for file management"
source="https://github.com/lxde/libfm/archive/${version}.tar.gz"
depends="gtk3,pango,intltool,menu-cache"
group="x11.libs"

setup(){
        cd $SOURCEDIR

        autoreconf -fvi
         ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --sysconfdir=/etc \
                --with-gtk=3 \
                --with-gnu-ld
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

menu-cache

LXDE için gerekli temel kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="menu-cache"
version="1.1.0"
description="Caching mechanism for freedesktop.org compliant menus"
source="https://github.com/lxde/menu-cache/archive/refs/tags/$version.tar.gz"
depends="libfm-extra"
group="lxde.base"

setup(){
        mkdir -p /tmp/bps/build/patches
        cp ${dizin}/${paket}/patches/* /tmp/bps/build/patches/
        cd $SOURCEDIR
        patch -Np1 < ../patches/menu-cache-1.1.0-0001-Support-gcc10-compilation.patch
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libfm-extra

LXDE için gerekli temel kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libfm-extra"
version="1.3.2"
description="Library for file management"
source="https://github.com/lxde/libfm/archive/refs/tags/${version}.tar.gz"
depends="gtk3,pango,intltool,menu-cache"
group="x11.libs"


setup(){
        mkdir -p /tmp/bps/build/patches
        cp ${dizin}/${paket}/patches/* /tmp/bps/build/patches/
        cd $SOURCEDIR
        patch -Np1 -i ../patches/libfm-fixes.patch
        autoreconf -fvi
         ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
                  --with-gtk=3   \
                --with-extra-only
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxmenu-data

LXDE için gerekli temel kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxmenu-data"
version="0.1.5"
description="Provides files needed for LXDE application menus"
source="https://github.com/lxde/lxmenu-data/archive/refs/tags/$version.tar.gz"
depends=""
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --sysconfdir=/etc
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxde-common

LXDE için gerekli temel kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxde-common"
version="0.99.2"
description="LXDE Session default configuration files and nuoveXT2 iconset "
source="https://github.com/lxde/lxde-common/archive/refs/tags/$version.tar.gz"
depends=""
builddepend=""
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxappearance

GTK temalarını yapılandırma aracıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxappearance"
version="0.6.3"
description="Feature-rich GTK+ theme switcher of the LXDE Desktop"
source="https://downloads.sourceforge.net/lxde/$name-$version.tar.xz"
depends="gtk3,dbus-glib"
builddepend=""
group="lxde.base"

setup(){
        cp -prvf $PACKAGEDIR/00-gtk3-fix.patch /tmp/bps/build/
        cd $SOURCEDIR
        patch -Np1 -i ../00-gtk3-fix.patch
        ./configure --prefix=/usr --enable-gtk3 \
            --libdir=/usr/lib64/ \
            --enable-dbus

}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxinput

Giriş aygıtlarını yapılandırma için gerekli bileşendir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxinput"
version="0.3.5"
description="LXDE keyboard and mouse configuration tool"
source="https://downloads.sourceforge.net/lxde/lxinput-$version.tar.xz"
depends="gtk3"
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --enable-gtk3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxrandr

Ekran çözünürlüğü ve monitör yapılandırması için kullanılan kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxrandr"
version="0.3.2"
description="Monitor configuration tool (part of LXDE)"
source="https://downloads.sourceforge.net/lxde/lxrandr-$version.tar.xz"
depends="gtk3,xrandr"
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
                --enable-gtk3 \
                --enable-man \
            --libdir=/usr/lib64/
}

build(){
        make DESTDIR=$DESTDIR
}

package(){
        make DESTDIR=$DESTDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxsession

LXDE oturum yöneticisi.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxsession"
version="0.5.5"
description="Lightweight X11 session manager"
#source="https://downloads.sourceforge.net/lxde/lxsession-$version.tar.xz"
source="https://salsa.debian.org/lxde-team/lxsession/-/archive/debian/$version-2/lxsession-debian-$version-2.tar.gz"
depends="gtk3,polkit"
builddepend="intltool,vala,docbook-xsl"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr\
            --sysconfdir=/etc \
            --libexecdir=/usr/libexec \
            --enable-gtk3 \
            --enable-buildin-clipboard \
            --libdir=/usr/lib64/ \
            --enable-buildin-polkit
}

build(){
        make
}

package(){
        make  DESTDIR="$DESTDIR" install
        # Ignore package by AppStream to avoid duplicated IDs
        echo "X-AppStream-Ignore=true" >> "$DESTDIR/usr/share/applications/lxsession-default-apps.desktop"
        echo "X-AppStream-Ignore=true" >> "$DESTDIR/usr/share/applications/lxsession-edit.desktop"
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxpanel

Lxde görev çubuğu ve panel bileşeni.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxpanel"
version="0.10.1"
description="Lightweight X11 desktop panel for LXDE"
source="https://downloads.sourceforge.net/lxde/lxpanel-$version.tar.xz"
depends="libwnck3,wireless-tools,curl,alsa-lib,libwnck,"
builddepend="intltool,docbook-xml,docbook-xsl,wireless_tools"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        # Disable pager plugin as it breaks panel layout with GTK+ 3
        # https://sourceforge.net/p/lxde/bugs/773/
        sed -i "/pager.c/d" plugins/Makefile.am
        sed -i "/STATIC_PAGER/d" src/private.h
        sed -i "s/libwnck-3.0//" configure.ac
        sed -i "s/background=1/background=0/" data/default/panels/panel.in

        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --enable-gtk3 \
            --disable-silent-rules \
            --with-plugins=all,-cpufreq
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxtask

Lxde görev yöneticisidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxtask"
version="0.1.10"
description="LXDE Task manager"
source="https://github.com/lxde/lxtask/archive/refs/tags/$version.tar.gz"
depends="gtk3"
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --enable-gtk3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxterminal

LXDE için terminal öykünücüsüdür.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxterminal"
version="0.4.0"
description="Lightweight vte-based tabbed terminal emulator for LXDE"
source="https://salsa.debian.org/lxde-team/lxterminal/-/archive/debian/$version-2/lxterminal-debian-$version-2.tar.gz"
#source="https://downloads.sourceforge.net/lxde/lxterminal-$version.tar.xz"
depends="vte3,gtk3,dejavu"
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure \
                --prefix=/usr \
            --libdir=/usr/lib64/ \
            --enable-gtk3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

pcmanfm

Dosya yöneticisi.

Paketi Derleme :

```
#!/usr/bin/env bash
name="pcmanfm"
version="1.3.2"
description="LXDE keyboard and mouse configuration tool"
source="https://github.com/lxde/pcmanfm/archive/refs/tags/$version.tar.gz"
depends="lxmenu-data,libfm,menu-cache,libkeybinder3"
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --with-gtk=3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxlauncher

Netbook tarzı uygulama başlatıcı.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxlauncher"
version="0.2.5"
description="Open source clone of the Asus launcher for EeePC"
source="https://github.com/lxde/lxlauncher/archive/refs/tags/$version.tar.gz"
depends="startup-notification,gtk3,menu-cache"
builddepend="intltool"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --sysconfdir=/etc \
            --enable-gtk3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxhotkey

Klavye kısayollarının yönetimi için kullanılır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxhotkey"
version="0.1.1"
description="LXDE hotkey"
source="https://github.com/lxde/lxhotkey/archive/refs/tags/$version.tar.gz"
depends="libfm"
builddepend="libexif"
group="lxde.base"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --with-gtk=3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lxde-icon-theme

Lxde ikon ve tema paketidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lxde-icon-theme"
version="0.5.1"
description="LXDE default icon theme based on nuoveXT2"
source="https://downloads.sourceforge.net/lxde/lxde-icon-theme-$version.tar.xz"
depends=""
builddepend=""
group="lxde.base"


setup(){
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libjpeg-turbo

JPEG görüntü kodek kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libjpeg-turbo"
version="3.0.3"
description="MMX, SSE, and SSE2 SIMD accelerated JPEG library"
source="https://github.com/libjpeg-turbo/libjpeg-turbo/archive/refs/tags/$version.tar.gz"
depends=""
group="media.libs"


setup(){
    cd $SOURCEDIR
    mkdir build
    cd build

    cmake -DCMAKE_INSTALL_PREFIX=/usr \
        -DCMAKE_INSTALL_LIBDIR=/usr/lib64 \
        -DWITH_JPEG8=ON ..
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
    mkdir -p $DESTDIR/lib64
    cd $DESTDIR/lib64
ln -s ../usr/lib64/libjpeg.so.8 libjpeg.so.8.2.2
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

cairo

Grafikleri çizimi için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="cairo"
version="1.18.0"
description="A vector graphics library with cross-device output support"
source="https://gitlab.freedesktop.org/cairo/cairo/-/archive/$version/cairo-$version.tar.gz"
depends="pixman,libpng,glib,libpcre2,fontconfig,libX11"
group="x11.libs"


setup(){
    cd $SOURCEDIR
    meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/ \
        -D b_lto=true \
        -D gtk_doc=false \
        -D spectre=disabled \
        -D symbol-lookup=disabled \
        -D tests=disabled
}

build(){
    ninja -C $BUILDDIR
}

package(){
    DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gdk-pixbuf

GTK+ kütüphanesinin bir parçası olan pakettir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gdk-pixbuf"
version="2.42.10"
description="gdk-pixbuf Image loading library for GTK+ "
source="https://download.gnome.org/sources/gdk-pixbuf/${version%.*}/gdk-pixbuf-$version.tar.xz"
depends="libjpeg-turbo,shared-mime-info,gi-docgen,libtiff,libpng,glib"
group="x11.libs"


setup(){
    cd $SOURCEDIR
    meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/ \
        -Db_lto=true \
        -D builtin_loaders=all \
        -Dman=false \
        -D installed_tests=false \
        -D tiff=enabled \
        -D introspection=enabled \
        -D tests=false

}

build(){
    meson compile -C $BUILDDIR
    #ninja -C $BUILDDIR
}

package(){
    DESTDIR="$DESTDIR" meson install -C $BUILDDIR

}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gtksourceview4

GTK+ tabanlı uygulamalar için metin düzenleme paketidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gtksourceview4"
version="4.8.4"
description="A text widget adding syntax highlighting and more to GNOME"
source="https://download.gnome.org/sources/gtksourceview/${version%.*}/gtksourceview-${version}.tar.xz"
depends="gtk3,libxml2,gobject-introspection"
group="x11.libs"


setup(){
    cd $SOURCEDIR
    meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/ \
        -Db_lto=true \
        -Dgtk_doc=true
}

build(){
    ninja -C $BUILDDIR
}

package(){
    DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

hsakmt

AMD ROCm için bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="hsakmt"
version="1.0.0"
description="Package hsakt"
source="https://www.x.org/archive/individual/lib/hsakmt-$version.tar.gz"
depends=""
group="x11.libs"

setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libFS

X Window System'de yazı tipi kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libFS"
version="1.0.9"
description="X.Org FS library"
source="https://www.x.org/archive/individual/lib/libFS-$version.tar.xz"
depends="xorgproto"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libnotify

Bildirimlerini göstermek için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libnotify"
version="0.8.2"
description="Library for sending desktop notifications"
source="https://download.gnome.org/sources/libnotify/${version%.*}/libnotify-${version}.tar.xz"
depends=""
group="x11.libs"

setup(){
    cd $SOURCEDIR
    meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/ \
        -Dgtk_doc=false     \
        -Dman=false \
            -Ddefault_library=both
}

build(){
    ninja -C $BUILDDIR
}

package(){
    DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libvdpau

NVIDIA'nın VDPAU arayüzünü için kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libvdpau"
version="1.5"
description="VDPAU wrapper and trace libraries"
source="https://gitlab.freedesktop.org/vdpau/libvdpau/-/archive/$version/libvdpau-$version.tar.gz"
depends="libXext"
group="x11.libs"


setup(){
    cd $SOURCEDIR
    meson setup $BUILDDIR --prefix=/usr \
        -Ddefault_library=both \
        --libdir=/usr/lib64 \
        -Ddri2=true
}

build(){
    ninja -C $BUILDDIR
}

package(){
    DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libwnck3

GNOME masaüstü ortamı için gerekli kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libwnck3"
version="43.0"
description="A window navigation construction kit "
source="https://gitlab.gnome.org/GNOME/libwnck/-/archive/43.0/libwnck-43.0.tar.gz"
depends="startup-notification"
group="x11.libs"


setup(){
    cd $SOURCEDIR
    meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    ninja -C $BUILDDIR
}

package(){
    DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXaw3d

Xaw için 3D kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXaw3d"
version="1.6.4"
description="X.Org Xaw3d library"
source="https://www.x.org/archive/individual/lib/libXaw3d-$version.tar.gz"
depends="libX11,libXt,libXmu,libXext"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXcomposite

Pencere düzeni için gerekli kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXcomposite"
version="0.4.6"
description="X.Org Xcomposite library"
source="https://gitlab.freedesktop.org/xorg/lib/libxcomposite/-/archive/libXcomposite-$version/libxcomposite-libXcomposite-$version.tar.gz"
depends=""
group="x11.libs"

setup(){
    cd $SOURCEDIR
    autoreconf -fvi
    ./configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXdamage

Pencerenin değişimini takip etmek kullanılır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXdamage"
version="1.1.6"
description="X.Org Xdamage library"
source="https://www.x.org/archive/individual/lib/libXdamage-$version.tar.xz"
depends="libXfixes,libX11"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXfont

Xorg font dosyalarını yönetmesi için grekeli kütüphane.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXfont"
version="1.5.4"
description="X.Org Xfixes library"
source="https://www.x.org/archive/individual/lib/libXfont-$version.tar.gz"
depends="libfontenc,freetype"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libxkbcommon

Klavye için gerekli kütüphane.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libxkbcommon"
version="1.7.0"
description="keymap handling library for toolkits and window systems"
source="https://github.com/xkbcommon/libxkbcommon/archive/refs/tags/xkbcommon-$version.tar.gz"
depends="libxml2,libxcb,xkeyboard-config,wayland-protocols,wayland"
makedepend="bison,wayland-protocols,wayland"
group="x11.libs"

setup(){
cd $SOURCEDIR
    meson setup $BUILDDIR \
        --prefix=/usr \
        --libdir=/usr/lib64/ \
        --libexecdir=/usr/lib64/ \
        -Denable-docs=false \
        -Denable-wayland=true
}

build(){
    ninja -C $BUILDDIR
}

package(){
    DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libxkbui

Klavye için gerekli bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libxkbui"
version="1.0.2"
description="Package libxkbui"
source="https://www.x.org/archive/individual/lib/libxkbui-$version.tar.gz"
depends="libX11,libXt,libxkbfile"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libxklavier

Klavye düzenlerini yapılandırması için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libxklavier"
version="5.4"
description="High-level API for X Keyboard Extension"
source="https://people.freedesktop.org/~svu/libxklavier-${version}.tar.bz2"
depends="glib,iso-codes,libxml2"
group="x11.libs"


setup(){
cd  $SOURCEDIR
    autoreconf -fvi
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/ \
            --with-xkb-bin-base=/usr/bin \
            --with-xkb-base=/usr/share/X11/xkb \
            --enable-gtk-doc
}

build(){
    make
}

package(){
#mkdir -p $DESTDIR/usr/share/X11/xkb
    make install DESTDIR=$DESTDIR $jobs
    mkdir -p $DESTDIR/usr/lib64/pkgconfig
    install $SOURCEDIR/libxklavier.pc $DESTDIR/usr/lib64/pkgconfig/libxklavier.pc

}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXpresent

X11 sistemlerinde “Present” (sunum) uzantısını kullanan kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXpresent"
version="1.0.1"
description="X Present Extension C Library"
source="https://www.x.org/archive/individual/lib/libXpresent-$version.tar.xz"
depends="libX11,xorgproto,libXext,libXfixes,libXrandr"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXres

Xorg kaynak kullanımını izlemek için kullanılır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXres"
version="1.2.2"
description="X.Org XRes library"
source="https://www.x.org/archive/individual/lib/libXres-$version.tar.xz"
depends="libX11,libXext,xorgproto"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libxss

Ekran koruyucusu kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libxss"
version="1.2.4"
description="X11 Screen Saver extension library"
source="https://xorg.freedesktop.org/releases/individual/lib/libXScrnSaver-${version}.tar.xz"
depends="libXext,util-macros,xorgproto"
group="x11.libs"



setup(){
    autoreconf -fvi
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXtst

Klavye ve fare simüle etmek için kullanılan kütüphane.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXtst"
version="1.2.4"
description="X.Org Xlib-based client API for the XTEST & RECORD extensions library"
source="https://www.x.org/archive/individual/lib/libXtst-$version.tar.xz"
depends="libX11,libXext,xorgproto,libXi"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXv

X11 pencere yönetimi için kullanılır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXv"
version="1.0.12"
description="X.Org Xv library"
source="https://www.x.org/archive/individual/lib/libXv-$version.tar.xz"
depends=""
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXvMC

Video kütüphanesi.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXvMC"
version="1.0.13"
description="X.Org XvMC library"
source="https://www.x.org/archive/individual/lib/libXvMC-$version.tar.xz"
depends="libX11,libXext,libXv,xorgproto"
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXxf86dga

Uygulamaların grafik donanımına erişmesini sağlar.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXxf86dga"
version="1.1.6"
description="X.Org Xxf86dga library"
source="https://www.x.org/archive/individual/lib/libXxf86dga-$version.tar.gz"
depends=""
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libXxf86vm

Ekran çözünürlüğü, yenileme hızı gibi işlemlerin değiştirmesini sağlar.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libXxf86vm"
version="1.1.5"
description="X.Org Xxf86vm library"
source="https://www.x.org/archive/individual/lib/libXxf86vm-$version.tar.xz"
depends=""
group="x11.libs"


setup(){
    $SOURCEDIR/configure --prefix=/usr \
        --libdir=/usr/lib64/
}

build(){
    make
}

package(){
    make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

tslib

Yardımcı kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="tslib"
version="1.22"
description="Touchscreen Access Library"
source="https://github.com/libts/tslib/releases/download/${version}/tslib-${version}.tar.xz"
depends=""
group="x11.libs"


setup () {
        autoreconf -fvi
        $SOURCEDIR/configure --prefix=/usr \
                --libdir=/usr/lib64/
}

build () {
        make
}

package () {
        make DESTDIR="$DESTDIR" install $jobs
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

vte3

GTK+ tabanlıdır. Terminal öykünmesi sağlamak için kullanılır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="vte3"
version="0.76.2"
description="vte Library providing a virtual terminal emulator widget"
source="https://gitlab.gnome.org/GNOME/vte/-/archive/${version}/vte-${version}.tar.gz"
depends="cairo,fribidi,gcc,gdk-pixbuf2,glib,glibc,gnutls,icu,lz4,pango,pcre2"
builddepend="at-spi2-core,gi-docgen,git,gobject-introspection,gperf,gtk3,meson,vala"
group="x11.libs"

setup(){
cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
                --libdir=/usr/lib64/ \
                -Db_lto=false \
                -Ddocs=false \
                -Dgtk3=true \
                -Dgtk4=false \
                -D_systemd=false
}

build(){
        ninja -C $BUILDDIR
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xapp

GTK tabanlı yardımcı kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xapp"
version="2.8.0"
description="Common library for X-Apps project"
source="https://github.com/linuxmint/xapp/archive/$version/xapp-${version}.tar.gz"
depends="libdbusmenu,libgnomekbd,vala,py3-pygobject,make,gobject-introspection"
group="x11.libs"


setup(){
        cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
                --libdir=/usr/lib64/
}


build(){
        ninja -C $BUILDDIR
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xcb-util-cursor

XCB kütüphanesinin bir uzantısıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xcb-util-cursor"
version="0.1.4"
description="X C-language Bindings sample implementations"
source="https://www.x.org/releases/individual/xcb/xcb-util-cursor-${version}.tar.xz"
depends="libxcb,xcb-util-image,xcb-util-renderutil,xorgproto"
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
                --libdir=/usr/lib64 \
                --enable-shared \
                --enable-static \
                --disable-devel-docs \
                --without-doxygen
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xcb-util-errors

XCB kütüphanesinin bir uzantısıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xcb-util-errors"
version="1.0.1"
description="Package xcb-util-errors"
source="https://www.x.org/archive/individual/lib/xcb-util-errors-$version.tar.xz"
depends=""
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xcb-util-image

XCB kütüphanesinin bir uzantısıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xcb-util-image"
version="0.4.1"
description="X C-language Bindings sample implementations"
source="https://www.x.org/releases/individual/xcb/xcb-util-image-${version}.tar.xz"
depends="libxcb,xcb-util,xorgproto,util-macros"
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64 \
        --enable-shared \
        --enable-static \
        --disable-devel-docs \
        --without-doxygen
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xcb-util-keysyms

XCB kütüphanesinin bir uzantısıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xcb-util-keysyms"
version="0.4.1"
description="X C-language Bindings sample implementations"
source="https://www.x.org/archive/individual/lib/xcb-util-keysyms-$version.tar.xz"
depends=""
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
                --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xcb-util-renderutil

XCB kütüphanesinin bir uzantısıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xcb-util-renderutil"
version="0.3.10"
description="X C-language Bindings sample implementations"
source="https://www.x.org/releases/individual/xcb/xcb-util-renderutil-${version}.tar.xz"
depends="libxcb,util-macros,xorgproto"
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64 \
                --enable-shared \
                --enable-static \
                --disable-devel-docs \
                --without-doxygen
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xcb-util-wm

XCB kütüphanesinin bir uzantısıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xcb-util-wm"
version="0.4.2"
description="X C-language Bindings sample implementations"
source="https://www.x.org/archive/individual/lib/xcb-util-wm-$version.tar.xz"
depends=""
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

xtrans

Pencere yöneticisi için yardımcı kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="xtrans"
version="1.5.0"
description="X.Org xtrans library"
source="https://www.x.org/archive//individual/lib/xtrans-$version.tar.gz"
depends=""
group="x11.libs"


setup(){

        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make DESTDIR=$DESTDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libkeybinder3

Klavye kısayollarını yönetmek için kullanılan kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libkeybinder3"
version="0.3.2"
description="Musical key detection library for digital audio"
source="https://github.com/kupferlauncher/keybinder/releases/download/keybinder-3.0-v$version/keybinder-3.0-$version.tar.gz"
depends="gtk3"
builddepend="gtk-doc,gobject-introspection"
group="dev.libs"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --with-gtk=3
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libexif

JPEG ve TIFF formatındaki dosyalar için bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libexif"
version="0.6.25"
description="Library to parse an EXIF file and read the data from those tags"
source="https://github.com/libexif/libexif/archive/refs/tags/v${version}.tar.gz"
depends=""
group="media.libs"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make DESTDIR="$DESTDIR" install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gtk3

GTK3, C dilinde yazılmış, görsel kullanıcı arayüzüdür. GNOME kütüphanesidir, masaüstü uygulamaları geliştirmek için kullanılır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gtk3"
version="3.24.49"
description="Gimp Toolkit"
source="https://gitlab.gnome.org/GNOME/gtk/-/archive/${version}/gtk-${version}.tar.gz"
depends="cairo,libXi,libXext,libXfixes,libXcursor,libXdamage,librsvg,at-spi2-core,shared-mime-info,\
gdk-pixbuf,libxkbcommon,libepoxy,colord,libcloudproviders,dconf,cantarell-fonts,libpng,\
libXcomposite,libXinerama,glib,libXrandr,pango,wayland"
builddepend="sassc,libsass"
group="gui.libs"

setup(){
        CFLAGS="$CFLAGS -O2" \
        CXXFLAGS="$CXXFLAGS -O2" \

        XDG_DATA_DIRS=/usr/share/
        cp -r ${dizin}/${paket}/files/ /tmp/kly/build/
        cd $SOURCEDIR
        patch -Np1 -i ../files/0001-Allow-disabling-legacy-Tracker-search.patch

        meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/ \
        -Db_lto=true \
        -Dman=true \
        -Dgtk_doc=true \
        -Dbroadway_backend=true \
        -Dtests=false \
        -Dwayland_backend=false

}

build(){
   ninja -C $BUILDDIR
   #meson compile -C build
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install

        install -Dm644 /dev/stdin "$DESTDIR/usr/share/gtk-3.0/settings.ini" <<END
        [Settings]
        gtk-icon-theme-name = Adwaita
        gtk-theme-name = Adwaita
        gtk-font-name = Cantarell 11
        END

        #DESTDIR="$DESTDIR" meson install -C build
        #mkdir -p ${DESTDIR}/etc/sysconf.d
        mkdir -p ${DESTDIR}/etc/profile.d
        echo "export GTK_OVERLAY_SCROLLING=0" > ${DESTDIR}/etc/profile.d/gtk3.sh
        echo "export GDK_CORE_DEVICE_EVENTS=1" >> ${DESTDIR}/etc/profile.d/gtk3.sh

}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

shared-mime-info

Dosya türlerini tanımlamak için kullanılan bir standarttır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="shared-mime-info"
version="2.4"
description="The Shared MIME-info Database specification"
source="https://gitlab.freedesktop.org/xdg/shared-mime-info/-/archive/${version}/shared-mime-info-${version}.tar.gz"
depends="libxml2"
group="x11.misc"

setup(){

        cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        ninja -C $BUILDDIR
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lz4

Sıkıştırma algoritması ve kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lz4"
version="1.9.4"
description="Extremely fast compression algorithm"
source="https://github.com/lz4/lz4/archive/refs/tags/v1.9.4.tar.gz"
depends=""
builddepend=""
group="app.arch"

setup(){
        echo ""
}

build(){
        cd $SOURCEDIR
        make PREFIX=/usr
}

package(){
        make install DESTDIR=$DESTDIR PREFIX=/usr
        mv ${DESTDIR}/usr/{lib,lib64}
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gnutls

Kripto işlemleri uygulamak için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gnutls"
version="3.8.0"
url="https://www.gnupg.org/ftp/gcrypt/gnutls"
description="The GnuTLS Transport Layer Security Library"
source="https://www.gnupg.org/ftp/gcrypt/gnutls/v${version%.*}/gnutls-${version}.tar.xz"
depends="nettle,p11-kit,unbound"
group="net.libs"
uses_extra="zstd brotli libidn2"


setup(){
cd $SOURCEDIR
        ./configure --prefix=/usr \
        --libdir=/usr/lib64 \
        --disable-guile \
        --enable-ssl3-support \
        --enable-openssl-compatibility \
        --with-included-unistring \
        --with-included-libtasn1\
        --with-zstd \
        --with-brotli \
        --with-libidn2
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lcms2

Görüntü işleme ve renk yönetimi için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="lcms2"
version="2.16"
url="https://netix.dl.sourceforge.net/project/lcms/lcms/2.14/lcms2-2.14.tar.gz"
description="Small-footprint color management engine, version 2"
source="https://netix.dl.sourceforge.net/project/lcms/lcms/${version}/lcms2-${version}.tar.gz"
depends="libtiff"
group="media.libs"

setup(){

        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

cups

Yazıcı yönetimi için kullanılan pakettir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="cups"
version="2.4.7"
description="The Common Unix Printing System "
source="https://github.com/OpenPrinting/cups/releases/download/v${version}/cups-${version}-source.tar.gz"
depends="openssl"
group="net.print"


setup(){
        cp -prvf $PACKAGEDIR/files/ /tmp/kly/build/
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
                --sysconfdir=/etc \
                --localstatedir=/var \
                --with-menudir=/usr/share/applications \
                --with-icondir=/usr/share/icons \
                --with-logdir=/var/log/cups \
                --with-docdir=/usr/share/cups \
                --with-rundir=/run/cups \
                --with-cupsd-file-perm=0755 \
                --with-cups-user=lp \
                --with-cups-group=lp \
                --with-system-groups=lpadmin \
                --with-domainsocket=/run/cups/cups.sock \
                --enable-libusb \
                --without-rcdir \
                --without-php \
                --disable-pam \
                --enable-raw-printing \
                --enable-dbus \
                --with-dbusdir=/usr/share/dbus-1 \
                --enable-libpaper \
                --enable-ssl=yes \
                --enable-gnutls \
                --disable-launchd \
                --disable-systemd
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
        mkdir -p "$DESTDIR"/etc/{passwd,group,init}.d
        install ../files/cups.groups "$DESTDIR"/etc/group.d/cups
        install ../files/cups.users "$DESTDIR"/etc/passwd.d/cups
        install ../files/cupsd.initd "$DESTDIR"/etc/init.d/cupsd

        mv $DESTDIR/usr/lib/* $DESTDIR/usr/lib64/
        rm -rf $DESTDIR/usr/lib
}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gdbm

Anahtar-değer çifti olarak depolayan bir veritabanı yönetim sistemidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gdbm"
version="1.23"
description="GNU dbm is a set of database routines that use extensible hashing"
source="https://ftp.gnu.org/gnu/gdbm/gdbm-$version.tar.gz"
depends=""
builddepend=""
group="dev.libs"

setup(){
        cd $SOURCEDIR
        ./configure --prefix=/usr \
                --libdir=/usr/lib64 \
                --enable-libgdbm-compat \
                --disable-largefile \
                --disable-dependency-tracking \
                --enable-fast-install
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

mpdecimal

MPDecimal, çok hassas sayısal hesaplamalar yapmak için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="mpdecimal"
version="4.0.0"
url="https://www.bytereef.org/mpdecimal/index.html"
description="Package for correctly-rounded arbitrary precision decimal floating point arithmetic"
source="https://www.bytereef.org/software/$name/releases/$name-$version.tar.gz"
depends=""
builddepend=""
group="net.db"

setup(){
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libgusb

USB cihazlarla etkileşim için kullanılan bir C kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libgusb"
version="0.4.9"
url="https://github.com/hughsie/libgusb"
description=" GObject wrapper for libusb "
source="https://github.com/hughsie/libgusb/archive/refs/tags/$version.tar.gz"
depends="vala,gi-docgen,mesa,libusb,json-glib"
group="dev.libs"

setup(){
cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        ninja -C $BUILDDIR
}

package(){
   DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libisl

Sayısal kümeler üzerinde işlem için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libisl"
version="0.26"
description="An Integer Set Library for the Polyhedral Model"
source="https://libisl.sourceforge.io/isl-${version}.tar.bz2"
depends="gmp"
builddepend=""
group="dev.libs"

setup(){
        cd $SOURCEDIR

        ./configure --prefix=/usr \
                --mandir=/usr/share/man \
                --infodir=/usr/share/info \
                --localstatedir=/var
}

build(){
        make
}

package(){
        make DESTDIR=$DESTDIR install
        install -dm755 "${DESTDIR}"/usr/share/gdb/python/auto-load/usr/lib
        mv "$DESTDIR"/usr/lib/*-gdb.py "$DESTDIR"/usr/share/gdb/python/auto-load/usr/lib/
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libmpc

Sayı işleme kütüphanesi.

Paketi Derleme :

```
#!/usr/bin/env bash
version="1.3.1"
name="libmpc"
depends="glibc,mpfr"
description="Library for the arithmetic of complex numbers with arbitrarily high precision"
source="https://ftp.gnu.org/gnu/mpc/mpc-$version.tar.gz"
groups="dev.libs"

setup()
{
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64
}
build()
{
        make
}
package()
{
        make install DESTDIR=${DESTDIR}
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

duktape

Gömülü sistemler için JavaScript motorudur.

Paketi Derleme :

```
#!/usr/bin/env bash
name="duktape"
version="2.7.0"
url="https://duktape.org"
description="Embeddable Javascript engine"
source="https://duktape.org/$name-$version.tar.xz"
group="dev.lang"


_make(){
        make -f Makefile.sharedlibrary INSTALL_PREFIX=/usr "$@"
}
setup()
{
cd $SOURCEDIR
        # tools/configure.py needs Python 2
        sed -i 's/^#undef DUK_USE_FASTINT$/#define DUK_USE_FASTINT/' src/duk_config.h

        # Add missing NEEDED on libm.so
        sed -i 's/duktape\.c/& -lm/' Makefile.sharedlibrary

}
build(){
        _make
}

package(){
        _make DESTDIR="$DESTDIR" install
        install -Dt "$DESTDIR/usr/share/licenses/$name" -m644 LICENSE.txt
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

atkmm

ATK, masaüstü ortamlarında erişilebilirlik desteği için kullanılan araçtır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="atkmm"
version="2.28.3"
url="https://www.gtkmm.org"
description="Atkmm is the official C++ interface for the ATK accessibility toolkit library."
source="https://download.gnome.org/sources/atkmm/${version%.*}/atkmm-${version}.tar.xz"
depends="at-spi2-core,glibmm"
group="dev.cpp"

setup () {
cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
        --libdir=/usr/lib64/
}

build () {
        ninja -C $BUILDDIR
}

package () {
        DESTDIR=${DESTDIR} ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

at-spi2-core

Erişilebilirlik araçları ve yazılımları arasında iletişimi sağlar.

Paketi Derleme :

```
#!/usr/bin/env bash
name="at-spi2-core"

version="2.50.1"
url="https://gitlab.gnome.org/GNOME/at-spi2-core"
description="GTK+ & GNOME Accessibility Toolkit"

source="https://gitlab.gnome.org/GNOME/at-spi2-core/-/archive/AT_SPI2_CORE_${version//./_}/at-spi2-core-AT_SPI2_CORE_${version//./_}.tar.gz"
depends="glib,libffi,libpcre2,libXtst"

setup(){
cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/

}

build(){
        ninja -C $BUILDDIR
}

package(){
   DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libtiff

TIFF dosyalarını işlemek için kullanılan kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libtiff"
version="4.6.0"
description="Tag Image File Format (TIFF) library"
source="https://gitlab.com/libtiff/libtiff/-/archive/v$version/libtiff-v$version.tar.gz"
depends=""
group="media.libs"

setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --disable-docs
}

build(){
        make
}

package(){
        make DESTDIR="${DESTDIR}" install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libepoxy

Grafik kütüphaneleri arası uyumluluk sağlayan bir C kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libepoxy"
version="1.5.10"
description="Library for handling OpenGL function pointer management"
source="https://github.com/anholt/libepoxy/archive/refs/tags/$version.tar.gz"
depends="libX11,mesa"
group="media.libs"


setup(){
        cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        ninja -C $BUILDDIR
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gobject-introspection

GObject için temel kütüphane.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gobject-introspection"
version="1.72.0"
description="Introspection system for GObject-based libraries"
source="https://gitlab.gnome.org/GNOME/gobject-introspection/-/archive/${version}/gobject-introspection-${version}.tar.gz"
depends="cairo,libtool,python,py3-setuptools"
builddepend="bison,flex,libffi,meson"
group="dev.libs"

setup(){
        cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/ \
            --wrap-mode=nodownload
}

build(){
        ninja -C $BUILDDIR
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

p11-kit

PKCS#11 modülleri için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="p11-kit"
version="0.25.5"
url="https://github.com/p11-glue/p11-kit"
description="Library for loading and sharing PKCS#11 modules"
source="https://github.com/p11-glue/p11-kit/releases/download/${version}/p11-kit-${version}.tar.xz"
depends="libtasn1,elfutils,libffi"
group="app.crypt"

setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64 \
            --with-libffi \
                --without-systemd \
                --with-trust-paths=/etc/ssl/certs/ca-certificates.crt
}

build(){
        make
}

package(){
   make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

nettle

Güvenlik ve şifreleme işlemleri için kullanılır.Güvenli veri iletimi ve şifreleme aracıdır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="nettle"
version="3.9"
url="https://www.lysator.liu.se/~nisse/nettle"
description="A low-level cryptographic library"
source="https://ftp.gnu.org/gnu/nettle/nettle-${version}.tar.gz"
depends="glibc"
group="dev.libs"


setup(){
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

desktop-file-utils

Masaüstü ortamlarıyla ilgili dosya işlemleri yapan yazılım paketidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="desktop-file-utils"
version="0.27"
description="Command line utilities for working with desktop entries"
source="https://www.freedesktop.org/software/desktop-file-utils/releases/desktop-file-utils-${version}.tar.xz"
depends="glib"
builddepend="git,meson"
group="dev.util"


setup(){
        cd $SOURCEDIR
        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        ninja -C build
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libidn2

Alan Adları için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libidn2"
version="2.3.8"
description="Free software implementation of IDNA2008, Punycode and TR46"
source="https://ftp.gnu.org/gnu/libidn/libidn2-${version}.tar.gz"
depends="libunistring"
group="net.dns"

setup(){
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64 \
            --disable-static \
            --disable-nls
}
build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

locale-tr

locale-tr, Türkçe dil ve yerel ayarları sistemde uygun şekilde sağlamak için kullanılan bir yerelleştirme dosyası ya da paketi olup, sistemdeki tarih, saat, dil, para birimi gibi birçok yerel ayarın Türkçe'ye uyumlu olmasını sağlar.

Paketi Derleme :

```
#!/usr/bin/env bash
name="locale-tr"
version="1.0"
description="loacale türkçeleştirme ayarı"
source=""
depends="glibc,tzdata"
group="sys.apps"


setup(){
        echo ""
}

build(){
        echo ""
}

package(){
umask 022
mkdir -p ${DESTDIR}/lib64
mkdir -p ${DESTDIR}/lib64/locale

mkdir -p ${DESTDIR}/etc
mkdir -p ${DESTDIR}/etc/profile.d
cp -prfv $PACKAGEDIR/files/locale.sh  ${DESTDIR}/etc/profile.d/locale.sh
cp -prfv $PACKAGEDIR/files/keyboard.sh  ${DESTDIR}/etc/profile.d/keyboard.sh
cp -prfv $PACKAGEDIR/files/locale.gen  ${DESTDIR}/etc/locale.gen
cp -prfv $PACKAGEDIR/files/profile  ${DESTDIR}/etc/profile
}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

hicolor-icon-theme

Masaüstü ortamları ve uygulamalar için bir temadır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="hicolor-icon-theme"
version="0.17"
description="Freedesktop.org Hicolor icon theme"
source="http://icon-theme.freedesktop.org/releases/hicolor-icon-theme-$version.tar.xz"
depends=""
group="x11.themes"


setup(){
        cd $SOURCEDIR
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

polkit

Kullanıcıların güvenli bir şekilde yetkilendirmek için kullanılan bir araçtır.

Paketi Derleme :

```
#!/usr/bin/env bash
name="polkit"
version="123"
url="https://gitlab.freedesktop.org/polkit/polkit"
description="Application development toolkit for controlling system-wide privileges"
source="https://gitlab.freedesktop.org/polkit/polkit/-/archive/$version/polkit-$version.tar.gz"
depends="duktape,expat,glib,pam,elogind"
builddepend="gobject-introspection,gtk-doc,meson"
group="sys.auth"

setup(){
cp -prfv $PACKAGEDIR/files /tmp/kly/build/
cd $SOURCEDIR

        meson setup $BUILDDIR --prefix=/usr \
            --libdir=/usr/lib64/ \
            -Db_lto=true \
            -Dsession_tracking="libelogind" \
            -Dsystemdsystemunitdir=/trash \
                -Dpam_prefix=/etc/pam.d \
                -Dpolkitd_user=polkitd

}

build(){
        ninja -C $BUILDDIR $jobs
}

package(){
        DESTDIR=$DESTDIR ninja -C $BUILDDIR install $jobs
        DESTDIR="$BUILDDIR/elogind/dest" meson install --no-rebuild -C elogind

        chown -R polkitd:polkitd $DESTDIR/etc/polkit-1/rules.d $DESTDIR/usr/share/polkit-1/rules.d
        chmod -R 700 $DESTDIR/etc/polkit-1/rules.d $DESTDIR/usr/share/polkit-1/rules.d
        chmod 4755 $DESTDIR/usr/lib/polkit-1/polkit-agent-helper-1
        chmod 4755 $DESTDIR/usr/bin/pkexec

        rm -rf $DESTDIR/garbage
        install -d $DESTDIR/etc/init.d

        install -Dm755 ../files/polkit.initd $DESTDIR/etc/init.d/polkit
        mkdir -p $DESTDIR/var
        mkdir -p $DESTDIR/var/empty
}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libjpeg62

JPEG formatındaki resimler için kullanılan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libjpeg62"
version="0"
description="MMX, SSE, and SSE2 SIMD accelerated JPEG library"
source="http://ftp.de.debian.org/debian/pool/main/libj/libjpeg6b/libjpeg6b_6b2.orig.tar.gz"
depends=""
group="media.libs"

setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/lib64/libjpeg62
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
        mkdir -p $DESTDIR/lib64
        cd $DESTDIR/lib64
        ln -s libjpeg62/lib/libjpeg.so.62 libjpeg.so.62
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

gpicview

gpicview, GNOME masaüstü ortamı için resim görüntüleyicisidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="gpicview"
version="0.3.1"
description="Lightweight image viewer"
source="https://github.com/lxde/gpicview/archive/refs/tags/$version.tar.gz"
depends="gtk3"
builddepend="intltool"
group="lxde.apps"


setup(){
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
                --enable-gtk3 \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libdmx

DMX protokolünü kütüphanesidir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libdmx"
version="1.1.4"
description="X.Org dmx library"
source="https://www.x.org/archive/individual/lib/libdmx-$version.tar.gz"
depends=""
group="x11.libs"


setup(){
        $SOURCEDIR/configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lightdm

LightDM, Linux sistemlerinde kullanılan bir giriş yöneticisidir (display manager).

Paketi Derleme :

Debian'da paketi derlemek için aşağıdaki paketlerin kurulu olması gerekir.

```
sudo apt install libgcrypt20-dev, libxklavier-dev
```

```
#!/usr/bin/env bash
name="lightdm"
version="1.32.0"
description="A lightweight display manager"
source="https://salsa.debian.org/xfce-extras-team/lightdm/-/archive/debian/$version-1/lightdm-debian-$version-1.tar.gz"
depends="libX11,pam,libxklavier,xorg-server,libxcb,glib,libgcrypt,libXdmcp,xmodmap,xrdb,libXi"
group="x11.misc"


setup(){
        cp -prvf $PACKAGEDIR/files/ /tmp/kly/build/
        cd $SOURCEDIR
        ./autogen.sh
        ./configure --prefix=/usr --sysconfdir=/etc --libdir=/usr/lib64/ --localstatedir=/var \
                --disable-tests --disable-gtk-doc --enable-yelp-tools --enable-qt5-base \
                --with-greeter-user=lightdm --with-greeter-session=lightdm-gtk-greeter
}

build(){
        make
}

package(){
        make install DESTDIR=$DESTDIR
        # create user and group
        # install service
        mkdir -p $DESTDIR/etc/init.d
        install ../files/lightdm.initd $DESTDIR/etc/init.d/zlightdm
        mkdir -p $DESTDIR/usr/bin/
        # generate lightdm-session file
        #echo "#!/bin/sh" > $DESTDIR/usr/bin/lightdm-session
        #echo 'exec /etc/X11/xinit/Xsession $@' >> $DESTDIR/usr/bin/lightdm-session
        install -Dm755 ../files/lightdm-session $DESTDIR/usr/bin/lightdm-session
        # fix pam config
        sed -i "s/systemd/elogind/g" $DESTDIR/etc/pam.d/*

         for level in boot default nonetwork shutdown sysinit ; do
        mkdir -p ${DESTDIR}/etc/runlevels/$level
        done
install ${DESTDIR}/etc/init.d/zlightdm ${DESTDIR}/etc/runlevels/default/zlightdm
}
```

Ek dosyaları indirmek için tıklayınız..

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

lightdm-gtk-greeter

lightdm-gtk-greeter, LightDM için kullanılan bir greeter (karşılama arayüzü) paketidir.

Paketi Derleme :

Debian'da paketi derlemek için aşağıdaki paketlerin kurulu olması gerekir.

```
sudo apt install liblightdm-gobject-dev, xfce4-dev-tools
```

```
#!/usr/bin/env bash
name="lightdm-gtk-greeter"
version="2.0.8"
description="Gtk based greeter for lightdm."
source="https://github.com/Xubuntu/lightdm-gtk-greeter/releases/download/lightdm-gtk-greeter-$version/lightdm-gtk-greeter-$version.tar.gz"
depends="gtk3,lightdm"
builddepend="exo,xfce4-dev-tools"
group="x11.misc"

setup () {
        cd $SOURCEDIR
        autoreconf -fvi
        ./configure --prefix=/usr \
                --libdir=/usr/lib64/ \
                --sysconfdir=/etc \
                --sbindir=/usr/bin \
                --with-libxklavier \
                --enable-kill-on-sigterm \
                --disable-libido \
                --disable-libindicator
}

build(){
        make
}

package(){
        make DESTDIR=$DESTDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libgpg-error

GnuPG ve libgcrypt için yardımcı kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libgpg-error"
version="1.47"
url="https://www.gnupg.org/"
description="Support library for libgcrypt"

source="https://www.gnupg.org/ftp/gcrypt/libgpg-error/${name}-${version}.tar.bz2"
depends="glibc"
group="dev.libs"

setup(){
cd $SOURCEDIR
        autoreconf -vfi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/
}

build(){
        make
}

package(){
        make DESTDIR=$DESTDIR install
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

libgcrypt

Şifreleme, imzalama, hash ve rastgele sayı üretimi gibi işlemleri sağlayan bir kütüphanedir.

Paketi Derleme :

```
#!/usr/bin/env bash
name="libgcrypt"
version="1.10.3"
url="https://www.gnupg.org"
description="General purpose cryptographic library based on the code from GnuPG"
source="https://gnupg.org/ftp/gcrypt/libgcrypt/libgcrypt-1.10.3.tar.gz"
depends="libgpg-error"


setup(){
cd $SOURCEDIR
sed -i "s:t-secmem::" tests/Makefile.am
sed -i "s:t-sexp::" tests/Makefile.am

        autoreconf -fvi
        ./configure --prefix=/usr \
            --libdir=/usr/lib64/ \
            --disable-padlock-support
}

build(){
        make $jobs
}

package(){
        make DESTDIR=${DESTDIR} install $jobs
}
```

Not: Burada verilen derleme talimatı(script) **kly Paket Sistemi**'ni kullanarak paketi derler ve oluştur. Oluşan paket(**.kly uzantılı dosya**) **kly Paket Sistemi** kullanılarak siteme yüklenebilir. **kly Paket Sistemiyle Paket Yapma** konusunu okumak için tıklayınız.

ISO Hazırlama

Dağıtıma özgü ISO oluşturmak için o dağıtımın paket sistemi kullanılır. Bu dokümanda anlatılan **kly** paket sistemi, önceki bölümde detaylıca açıklanmıştır.

Bu bölümde, **kly** paket sistemini kullanarak ISO oluşturma adımları basit ve anlaşılır şekilde verilecektir. Aşağıda payşlaşılan script ile hazırlanan iso **/tmp/distro/distro.iso** konumunda olacaktır. Sistemi oluşturan paketleri;

- **github.com/kendilinuxunuyap/kly-base-packages**
- **github.com/kendilinuxunuyap/kly-x11-packages**
- **github.com/kendilinuxunuyap/kly-lxde-packages** konumlarından indirerek oluşturmaktadır.

Not: Anlatılan yöntem genel Linux dağıtımlarının ISO oluşturma mantığına dayanır. Araçlar, dizinler ve komutlarda küçük farklılıklar olabilir, ancak iş akışı aynıdır.

Aşağıdaki scriptle oluşturulmuş iso https://github.com/kendilinuxunuyap/kly-lxde-distro/releases/download/current/kly-x11-distro.iso adresinde bulunmaktadır.

kly Paket Sistemiyle ISO Oluşturma Scripti

```
#-------------------------------------------------------------------------------------------------------------------------
#!/bin/bash
set -x
if which apt &>/dev/null && [[ -d /var/lib/dpkg && -d /etc/apt ]] ; then
        apt-get update
        echo "işlem başladı....."
        apt install xorriso grub-pc-bin grub-efi mtools make python3 dosfstools e2fsprogs squashfs-tools \
        python3-yaml gcc wget curl unzip xz-utils debootstrap git erofs-utils zstd -y
fi
apt-get install git devscripts equivs -y
#-------------------------------------------------------------------------------------------------------------------------
rootfs="/tmp/distro/rootfs"
distro="/tmp/distro"
rm -rf $distro/iso
#rm -rf $rootfs
mkdir -p /tmp/distro/rootfs
mkdir -p $rootfs/bin
#mkdir -p distro/rootfs
#export PATH=/bbin:$PATH
cp busybox $rootfs/bin/busybox
cp kly $rootfs/bin/kly
cd $rootfs/bin/
ln -s busybox cpio
#busybox --install -s ./
cd $rootfs/
#cd distro/rootfs/
mkdir  -p var run dev sys proc etc tmp/kly/kur
bash -c "echo '127.0.0.1 kly' >> $rootfs/etc/hosts"
bash -c "echo 'kly' > $rootfs/etc/hostname"
bash -c "echo 'nameserver 8.8.8.8' > $rootfs/etc/resolv.conf"

### system chroot  bind/mount
for i in dev dev/pts proc sys; do mount -o bind /$i $rootfs/$i; done
### manuel kly-tools install
$rootfs/bin/busybox wget -O $rootfs/tmp/base-file-1.0.kly https://github.com/bpslinux/\
kly-temel-paketler/raw/refs/heads/master/base-file/base-file-1.0.kly 1>$rootfs/dev/null 2>$rootfs/dev/null
$rootfs/bin/busybox tar -xf $rootfs/tmp/base-file-1.0.kly -C $rootfs/tmp/kly/kur
$rootfs/bin/busybox tar -xf $rootfs/tmp/kly/kur/rootfs.tar.xz -C $rootfs

#paket adresi ekleniyor
$rootfs/bin/busybox mkdir -p $rootfs/etc/kly
echo "kendilinuxunuyap/kly-base-packages">$rootfs/etc/kly/sources.list
echo "kendilinuxunuyap/kly-x11-packages">>$rootfs/etc/kly/sources.list
echo "kendilinuxunuyap/kly-lxde-packages">>$rootfs/etc/kly/sources.list

### installing kly package in rootfs
$rootfs/bin/kly -u $rootfs
#**************************************************************************
echo root:x:0:0:root:/root:/bin/sh > $rootfs/etc/passwd
chmod 755 $rootfs/etc/passwd
#**************************************************************************

for paket in glibc readline ncurses bash openssl acl attr libcap libpcre2 gmp coreutils sqlite  \
util-linux grep sed mpfr gawk findutils libgcc libcap-ng gzip xz-utils zstd bzip2 elfutils libselinux tar zlib \
brotli curl
do
$rootfs/bin/kly -ri $paket $rootfs
#chroot $rootfs /bin/kly -ri $paket;
done

for paket in shadow file eudev cpio libsepol kmod audit libxcrypt libnsl libbsd libtirpc \
e2fsprogs dosfstools initramfs-tools libxml2 expat libmd libaio lvm2 popt icu iproute2 \
net-tools dhcp openrc rsync kbd kernel dialog live-boot live-config parted busybox nano grub \
efibootmgr efivar libssh openssh pam
do
#$rootfs/bin/kly -i $paket $rootfs
chroot $rootfs /bin/kly -ri $paket;
done

# burası x11 için olan paketlerdi. arasınsa ek paket varmı kontrol etmedim. edeceğim
for paket in xorg-server pixman libpciaccess libXau libXdmcp libXfont2 libxshmfence libdrm libxcvt libfontenc \
freetype libpng harfbuzz glib xterm libXft fontconfig libXext libXaw libXmu libXinerama libXpm libXt libX11 libICE \
libXrender libxcb libSM xf86-input-libinput xf86-input-vmmouse xf86-video-amdgpu xf86-video-ast xf86-video-ati \
xf86-video-dummy xf86-video-fbdev xf86-video-intel xf86-video-mga xf86-video-nouveau xf86-video-qxl xf86-video-r128 \
xf86-video-siliconmotion xf86-video-vboxvideo xf86-video-vesa xkbcomp libxkbfile libglvnd mesa xkeyboard-config \
libinput mtdev libevdev libwacom libgudev libffi xinit xcalc libXi openbox libXcursor libXfixes pango libXrandr \
fribidi xcb-util libthai libdatrie startup-notification dejavu libunwind dbus polkit elogind
do
chroot $rootfs /bin/kly -ri $paket;
done
```

```
#-------------------------------------------------------------------------------------------------------------------------
# scriptin devamı
# aşağıdaki paketleri kurunca lxsession açılıyor
for paket in libjpeg-turbo  cairo gdk-pixbuf gtksourceview4 hsakmt libdmx libfm libFS libnotify libvdpau libwnck3 \
libXaw3d libXcomposite libXdamage libXfont libxkbcommon libxkbui libxklavier libXpresent libXres libxss libXtst \
libXv libXvMC   libXxf86dga libXxf86vm tslib vte3 xapp xcb-util-cursor xcb-util-errors xcb-util-image \
xcb-util-keysyms xcb-util-renderutil xcb-util-wm xtrans
do
chroot $rootfs /bin/kly -ri $paket;
done

for paket in libfm-extra menu-cache lxappearance lxde-common lxde-icon-theme lxhotkey lxinput lxlauncher \
lxmenu-data lxpanel lxrandr lxsession lxtask lxterminal pcmanfm libkeybinder3 libexif gtk3 shared-mime-info \
lz4 gnutls lcms2 cups gdbm mpdecimal libgusb libisl libmpc duktape locale-tr
do
chroot $rootfs /bin/kly -ri $paket;
done

for paket in atkmm at-spi2-core libtiff libepoxy gobject-introspection hicolor-icon-theme p11-kit nettle \
desktop-file-utils libidn2
do
chroot $rootfs /bin/kly -ri $paket;
done

### user add and passwd
chroot $rootfs echo -e "1\n1\n"|chroot $rootfs passwd root
chroot $rootfs useradd live -m -s /bin/sh -d /home/live
chroot $rootfs echo -e "live\nlive\n"|chroot $rootfs passwd live
for grp in users tty wheel cdrom audio dip video plugdev netdev; do
chroot $rootfs usermod -aG $grp live || true
done

### update-initrd
fname=$(basename $rootfs/boot/config*)
kversion=${fname:7}
mv $rootfs/boot/config* $rootfs/boot/config-$kversion
cp $rootfs/boot/config-$kversion $rootfs/etc/kernel-config
chroot $rootfs update-initramfs -u -k $kversion

#### system chroot umount
for dir in dev dev/pts proc sys ; do    while umount -lf -R $rootfs/$dir 2>/dev/null ; do true; done done

#************************iso *********************************
mkdir -p $distro/iso
mkdir -p $distro/iso/boot
mkdir -p $distro/iso/boot/grub
mkdir -p $distro/iso/live || true

#### Copy kernel and initramfs (Debian/Devuan)
cp -pf $rootfs/boot/initrd.img-* $distro/iso/boot/initrd.img
cp -pf $rootfs/boot/vmlinuz-* $distro/iso/boot/vmlinuz
rm -rf $rootfs/boot
#### Create squashfs
mksquashfs $rootfs $distro/filesystem.squashfs -comp xz -wildcards
mv $distro/filesystem.squashfs $distro/iso/live/filesystem.squashfs

# grub.cfg dosyasını yaz
cat << EOF > "$distro/iso/boot/grub/grub.cfg"
set timeout=6; set default=1; terminal_input console;
menuentry "Canli(live) GNU/Linux 64-bit" --class liveiso {
linux /boot/vmlinuz boot=live init=/sbin/openrc-init net.ifnames=0 \
biosdevname=0
initrd /boot/initrd.img
}
menuentry "Kur GNU/Linux 64-bit" --class liveiso {
linux /boot/vmlinuz boot=live init=/bin/kur quiet
initrd /boot/initrd.img
}
EOF
#iso oluşturuluyor
grub-mkrescue $distro/iso/ -o $distro/distro.iso
```

Kaynaklar:

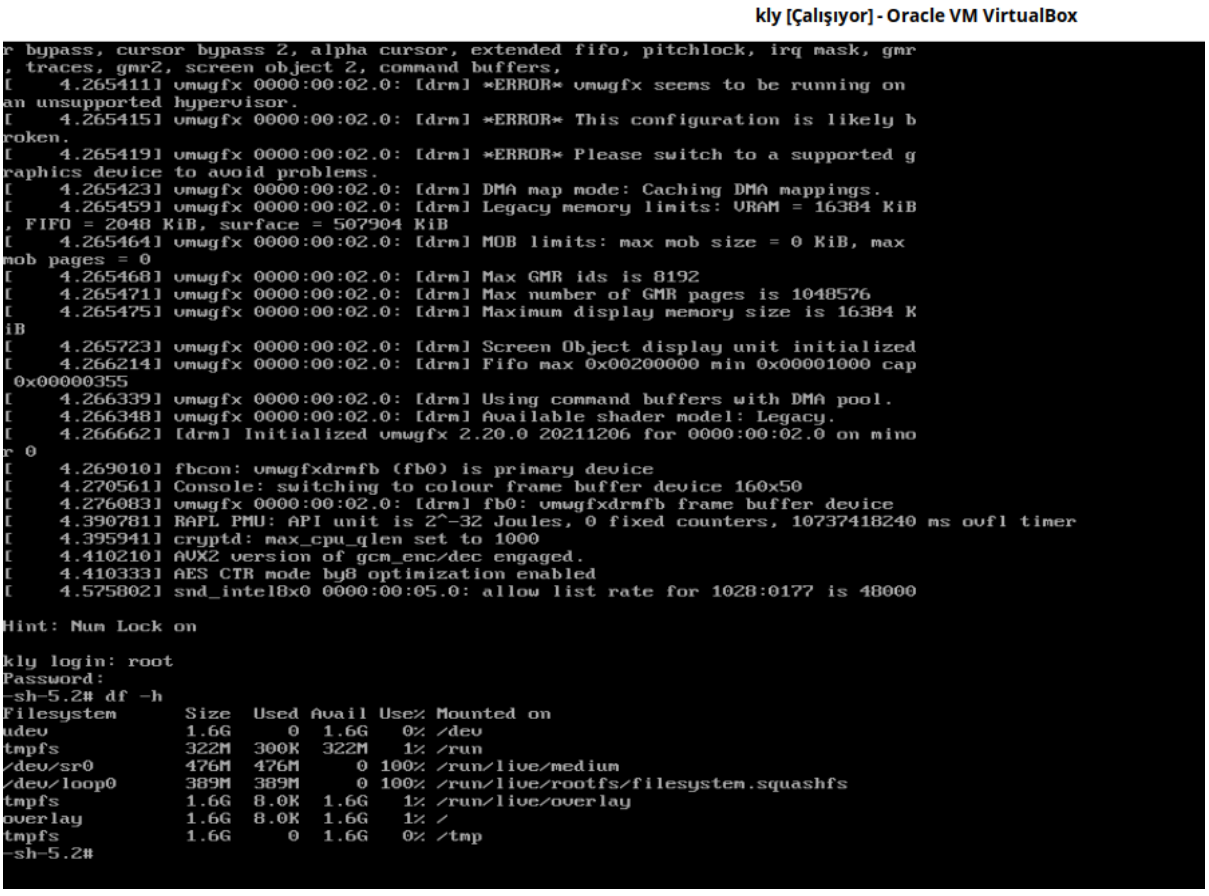

```
- https://www.subrat.info/build-kernel-and-userspace/ - 08/07/2025
- https://medium.com/@chienhaotan/compiling-and-running-a-minimal-kernel-with-busybox-bfc45a991017 - 08/07/2025
- https://wiki.archlinux.org/title/GRUB_(T%C3%BCrk%C3%A7e)  - 08/07/2025
- https://chatgpt.com/share/6875084c-6050-8012-9229-a37b47351aa2 - 14/07/2025
```

Oluşan Sistemin Çalıştırılması ve İncelenmesi

Bu dokümanda anlatılan paketlerin **kly Paket Sistemiyle** hazırlanmış **LXDE Masaüstü Ortamı** isosu hazırlandı. İsoyu https://github.com/kendilinuxunuyap/kly-lxde-distro/releases/download/current/kly-lxde-distro.iso adresinden indirebilirsiniz.

Şimdi hazırlanan ISO'yu **QEMU** veya **VirtualBox** kullanarak çalıştıralım. Ekran görüntüleri aşağıda verilmiştir.

Canlı(live) Sistem Kullanımı

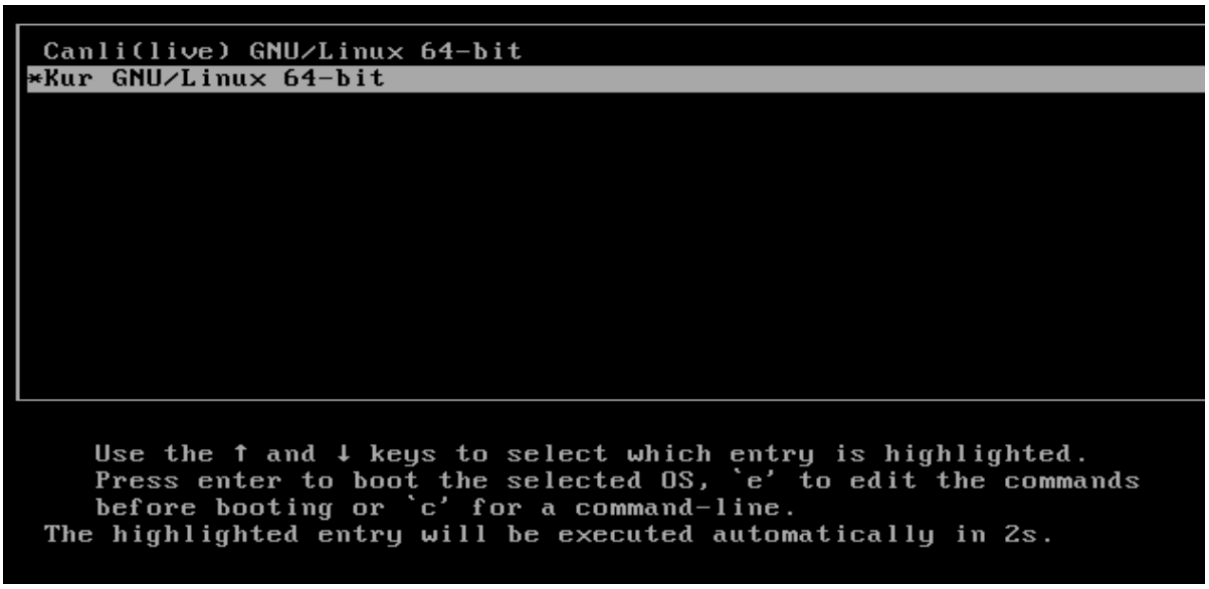


Canlı(live) sistem çalıştırıldığında overlay live bir sistemin açıldığını görmekteyiz. Canlı(live) sistemde kullanıcı adları ve parolaları;

- Kullanıcı: root Parola: 1
- Kullanıcı: live Parola: live

Sistem Kurulumu

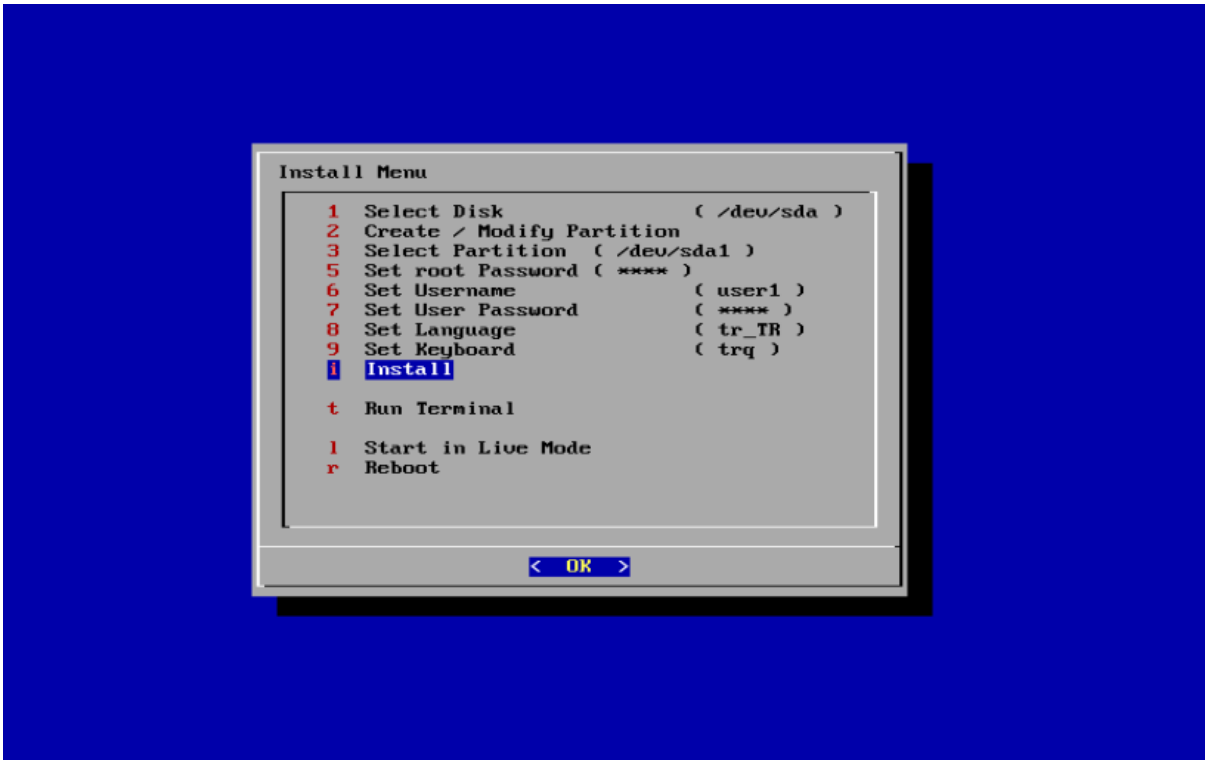
Sistemin kurulumu için resimlerde görünen sıraya göre seçimler yapmalıyız.



Kurulum menüsünde kullanıcı adları ve parolaları, klayve varsayılan olarak;

- Kullanıcı: root Parola: 1
- Kullanıcı: user1 Parola: 1
- Dil : tr_TR
- Klavye : trq

menüden değişiklik yapabilirsiniz. Değişiklik yapmadan sadece kurulum diskini ve disk bölümünü seçip Install(Yükle) işlemi yapabilirsiniz.

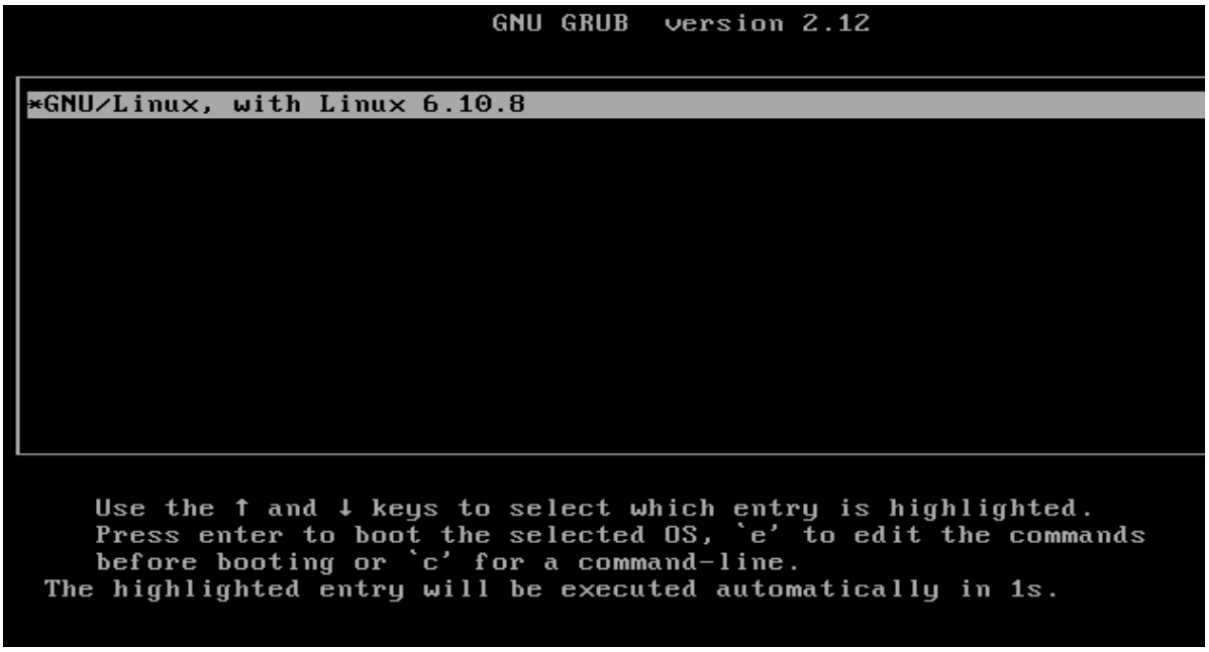


kly [Çalışıyor] - Oracle VM VirtualBox

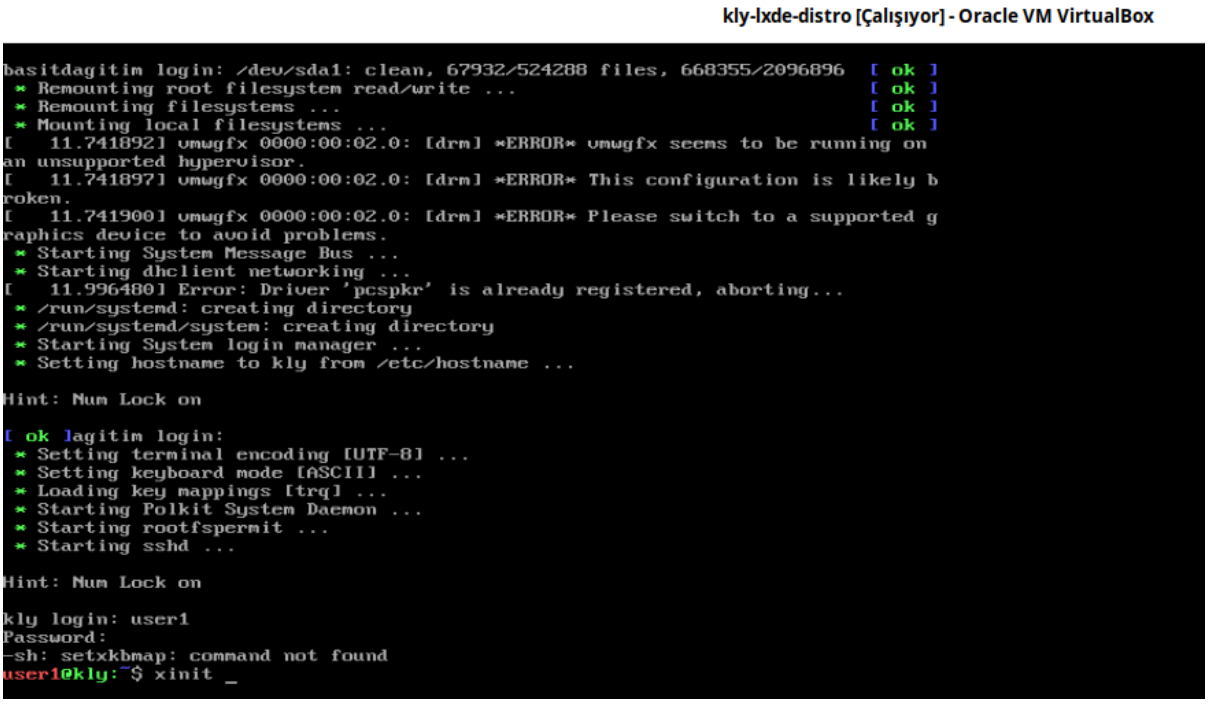
```
kuruluma geçildi........
legacy kurulum ........
Kurulum Yap■lacak Disk sda
mount: /kaynak: WARNING: source write-protected, mounted read-only.
**********disk bölümleri biçimlendiriliyor **********
e2fsck 1.47.0 (5-Feb-2023)
e2fsck: need terminal for interactive repairs
tune2fs 1.47.0 (5-Feb-2023)
Recovering journal.

This operation requires a freshly checked filesystem.

Please run e2fsck -f on the filesystem.

mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 2096896 4k blocks and 524288 inodes
Filesystem UUID: 9ea082d0-a034-4dab-8e2f-564e68c99c9c
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

Information: You may need to update /etc/fstab.

********** kurulum ba■lad■**********
Sistem yuklenmeye baslandi. Tahmini 3-5 dakika surecektir.. Lutfen bekleyiniz.....
.....
_
```

Sistemin Çalışması

Sistem kurulumu gerçekleştiğinde sistem resimde görüldüğü gibi açılmalıdır.



Sisteme **user1** (parola=1) kullanıcısı olarak giriş yapıldığı görülmektedir.

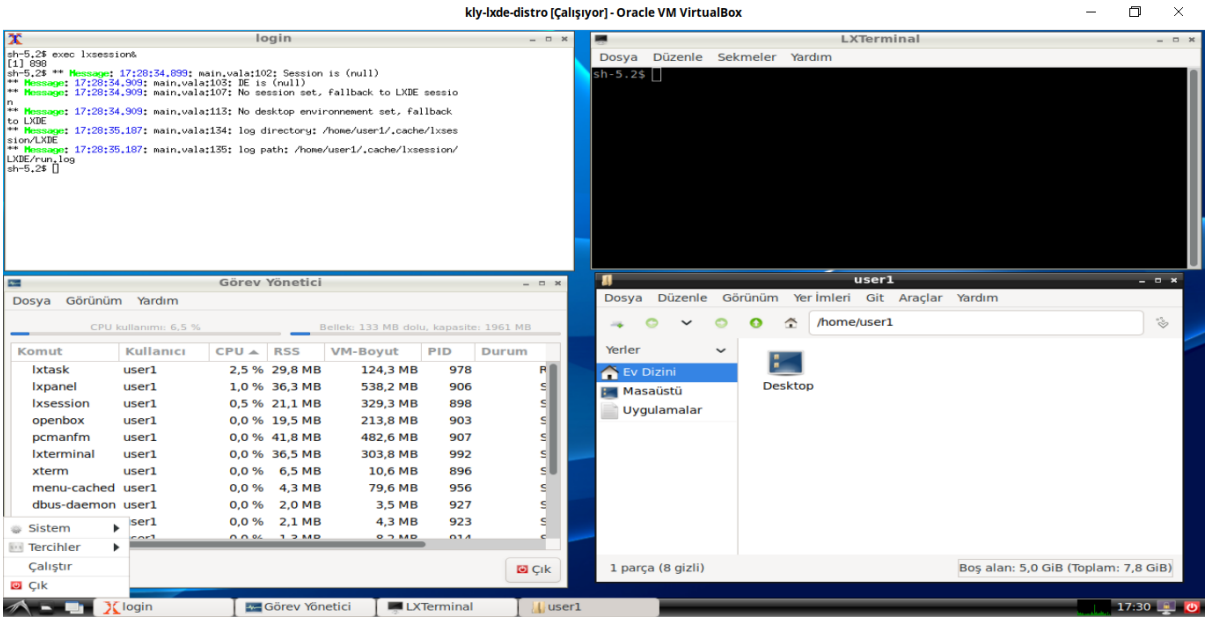


xinit çalışınca **X Pencere Sistemine** geçiş yapılacak.

xinit çalışınca **X Pencere Sistemi** aşağıdaki gibi açılacaktır.

X Pencere Sistemi çalışınca aşağı ve pencere yönetimini ve masaüstü ortamı için **exec lxsession** çalıştırılıyor.

Artık LXDE Masaüstü ortamındayız. LXDE araçlarının çalıştırılmış hali aşağıda görülmektedir.



LXDE, sade sistem isteyen kullanıcılar için mükemmel bir tercihtir. Ancak görsellik, modernlik veya geniş özellik seti beklentiniz varsa daha güncel alternatifler (XFCE, Cinnamon) tercih edilebilir.

Lxde Masaüstü Ortamı Çalıştırma

LXDE'yi xinit veya startx'le Çalıştırma

xinit ve startx X Pencere Sistemi'ni başlatmak için kullanılan komutlardır. `xinit` ve `startx`, X sunucusunu kullanıcı girişinden sonra manuel olarak başlatmak için kullanılır.

xinit X sunucusunu başlatır ve ardından belirtilen tek bir uygulamayı çalıştırır. **startx** `xinit` için bir kabuk (wrapper) komutudur. `~/.xinitrc` betiğini çalıştırır, yoksa sistem varsayılanını kullanır.

Eğer belirli bir pencere yöneticisi veya masaüstü ortamı ile başlatmak istiyorsanız, ~/.xinitrc dosyasını düzenlemeniz gerekebilir. Dosyanın içeriği şöyle olmalıdır:

openbox Masaüstü Ortamını Kullanma

```
exec openbox-session
```

lxde Masaüstü Ortamını Kullanma

```
exec lxsession
```

```
#startx
# veya
xinit
```

Lightdm

xinit veya **startx** ile masaüstü ortamının açılışını sağlanacağını bu bölümde gördük. Fakat bu şekilde bir açılış süreci yorucu ve mevcut dağıtımların açılışından farklı olacaktır. Bu durum son kullanıcı için çok çekici veya kullanıcı dostu gelmeyebilir. Bundan dolayı **lightdm** kullanabilirsiniz. LightDM bir Display Manager (giriş yöneticisi)'dir.

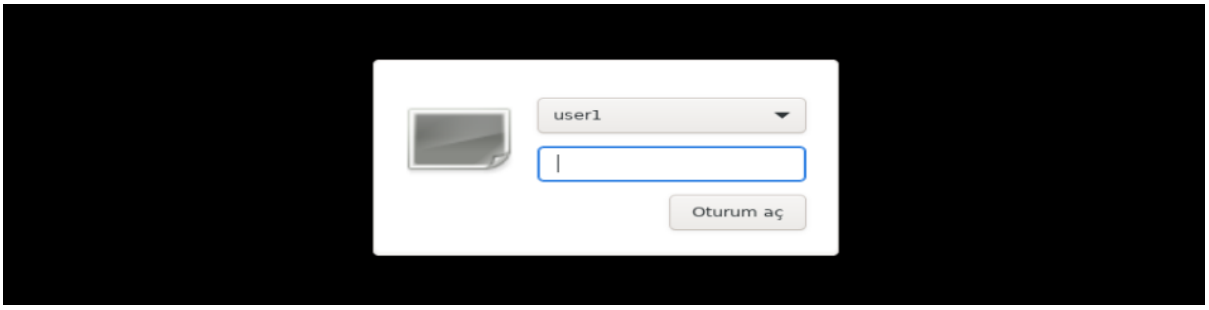
Görevi nedir?:

```
- Grafik arayüzlü giriş ekranını (login screen, greeter) sağlar.
- Kullanıcı adı/şifre sorar ve oturumu başlatır.
- Xorg veya Wayland oturumunu hazırlar.
- Kullanıcıya masaüstü ortamını (LXDE, XFCE, MATE vb.) seçme imkânı verebilir.
- Sistem açıldığında otomatik giriş (autologin) desteği vardır.
- Yani, LightDM = kullanıcı ile X sunucusu arasındaki köprüdür.
```

Mevcut sistemimize ek olarak derlediğimiz paketleri aşağıdaki komutlarla sisteminize kurabilirsiniz.

```
kly -ri lightdm
kly -ri libgcrypt
kly -ri libgpg-error
kly -ri lightdm-gtk-greeter
```

Bu paketler kurulduğunda aşağıda görüldüğü gibi açılışta ekran karşılayacak ve giriş yapılabilecektir.



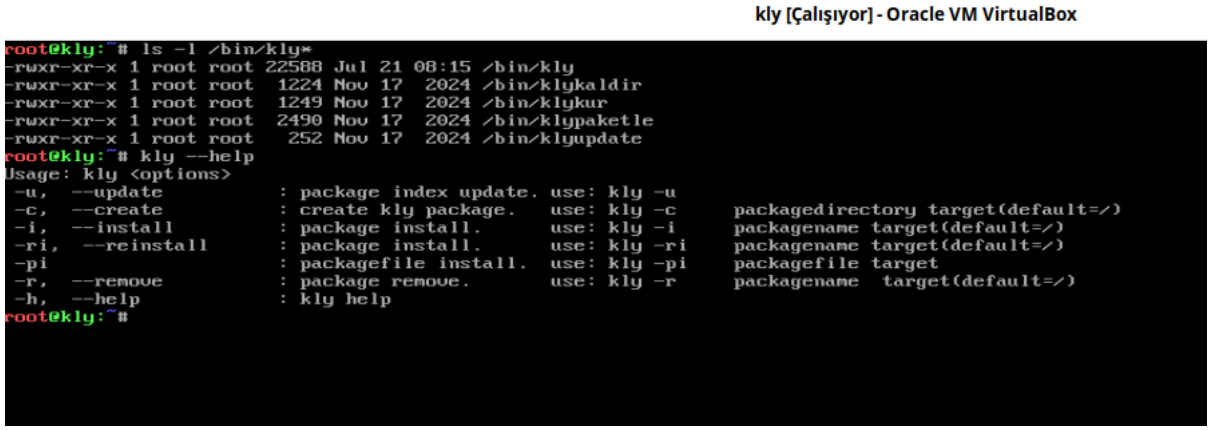
Listeye gelen kullanıcılardan birini seçerek porala doğrulamasından sonra masaüstü ortamı açılacaktır.

Yardımcı Konular

kly Paket Sistemi Kullanımı

Paket sistemi, sisteme paket **(kurma, kaldırma, index güncelleme)** gibi temel işlemlerin yapılmasını sağlar. **Temel Sistem** kitabımımızda **Paket Sistemi(kly)** tasarımı anlatılmıştı. Bu kitapta ise paketleri derlerken kly paket sistemi kullanılarak derleyeceğiz. Paket sistemi paketleri tekrar tekrar derlemek yerine paket olarak hazırlamakta ve istediğimiz zaman oluşturulan sisteme yüklenmekte veya kaldırılmatadır. Bir sorun olduğu farkedildiğinde ise tekrar derlenerek sadece bu paket günecelenebilmektedir. Bu tür avantajlardan dolayı paket sistemini kullanacağız.

Burada paket sisteminin nasıl kullanılacağı anlatılacaktır. Paket sistemimiz **Temel Sistem** üzerinde **base-file** paketiyle sisteme yüklendi. Aşağıda **kly Temel Sistem** üzerinde hazır gelen paket sistemi uygulamalarımız görülmektedir.



kly: *klypaketle, klykur, klykadir, klyupdate* scritplerimizin hepsini tek scriptte toplanmış halidir. Ayrıca paketlerin kurulumu ve kaldırılması sırasında bağımlılıklarınıda paketle birlikte kurmakta veya kaldırmakta. Bu dokümanda paketlerin derlenmesi, kurulması, kaldırılması vb. işlemler **kly** scripti kullanılarak yapılacaktır. **klypaketle, klypaketle, klykur, klykaldir, klyupdate** scripleri bir öğretici olması nedeniyle bulunmaktadır. Daha kapsamlı işlemler için yetersiz kalacaktır.

kly Paket Siteminin Debian'a Kurulumu

kly paket sistemi **Temel Sistem** üzerinde kurulu olarak gelmektedir. Üstte verilen resimde görülmektedir. Kullanımı ve parametrelerini unutmanız durumunda **kly --help** komutuyla kullanımı öğrenebilirsiniz.

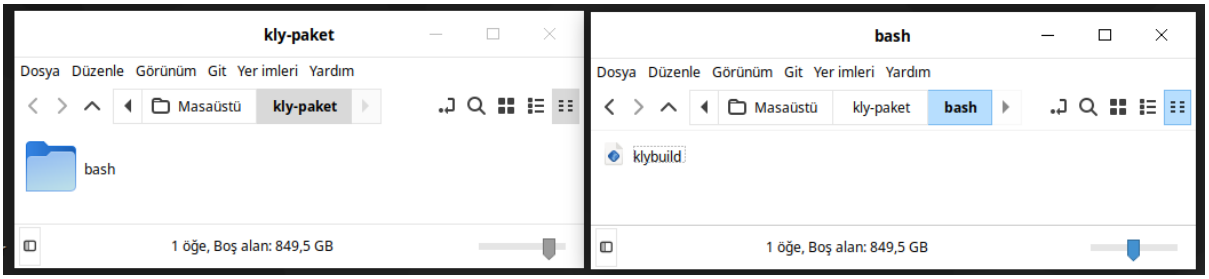
Debian üzerinde kullanmak için kly scriptimizi indirip **/bin** konumuna kopyalayalım. Aşağıda verilen komutları sırasıyla çalıştırınız. **kly** paket sisteminin sorunsuz çalışması için **busybox** paketinin kurulu olması gerekektedir. Sorunsuz çalışmışsa **Debian** sistemimize **kly** paket sistemimiz kurulmuştur.

```
wget https://kendilinuxunuyap.github.io/_static/files/base-file/kly
sudo mv kly /bin/kly
sudo chown root:root /bin/kly
sudo chmod 755 /bin/kly
sudo apt update
sudo apt install busybox-static -y
```

kly Paket Sistemiyle Paket Yapma

kly paket sistemi ile paket yapma işlemini Debian ortamında yapacağız. Debian üzerinde paket sistemimizi oluşturan scriptimiz **/bin/kly** konumunda olması gerekmektedir.

Şimdi **bash** paketinin **kly Paket Sistemi**'ni kullanarak derlemesini yapalım. Paket için aşağıda görüldüğü gibi masaüstüne **kly-paket** dizini oluşturuldu. **kly-paket** dizini içine **bash** dizini oluşturuldu. **bash** dizini içine **klybuild** dosyası oluşturuldu.



klybuild dosyasının içerine aşağıdaki kodu ekleyiniz.

```
#!/usr/bin/env bash
version="5.2.21"
name="bash"
depends="glibc,readline,ncurses"
description="GNU/Linux dağıtımında ön tanımlı kabuk"
source="https://ftp.gnu.org/pub/gnu/bash/${name}-${version}.tar.gz"
groups="app.shell"
setup() {
        cd $SOURCEDIR
        ./configure --prefix=/usr --libdir=/usr/lib64   --bindir=/bin \
                --with-curses --enable-readline --without-bash-malloc
}
build() {
        make
}
package()       {
        make install DESTDIR=$DESTDIR
        cd $DESTDIR/bin
        ln -s bash sh
}
```

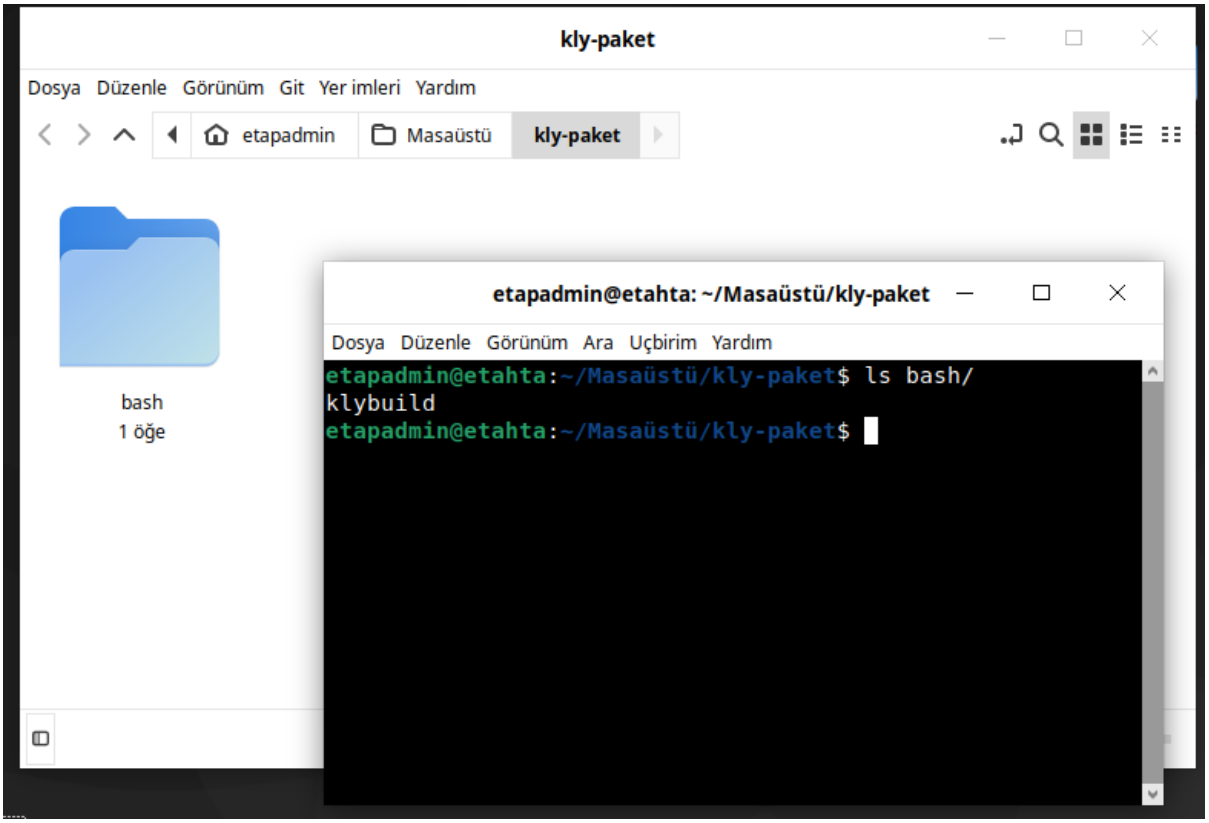
klybild dosyalarında Kullanılan Değişkenler

- **ROOTBUILDDIR:** /home/$user/distro/build → Derleme dizini
- **BUILDDIR:** /home/$user/distro/build/build-${name}-${version} → Paket derleme dizini
- **DESTDIR:** /home/$user/distro/rootfs → Yükleme dizini
- **PACKAGEDIR:** $(pwd) → Derleme scriptinin bulunduğu dizin
- **SOURCEDIR:** /home/$user/distro/build/${name}-${version} → Kaynak dizin

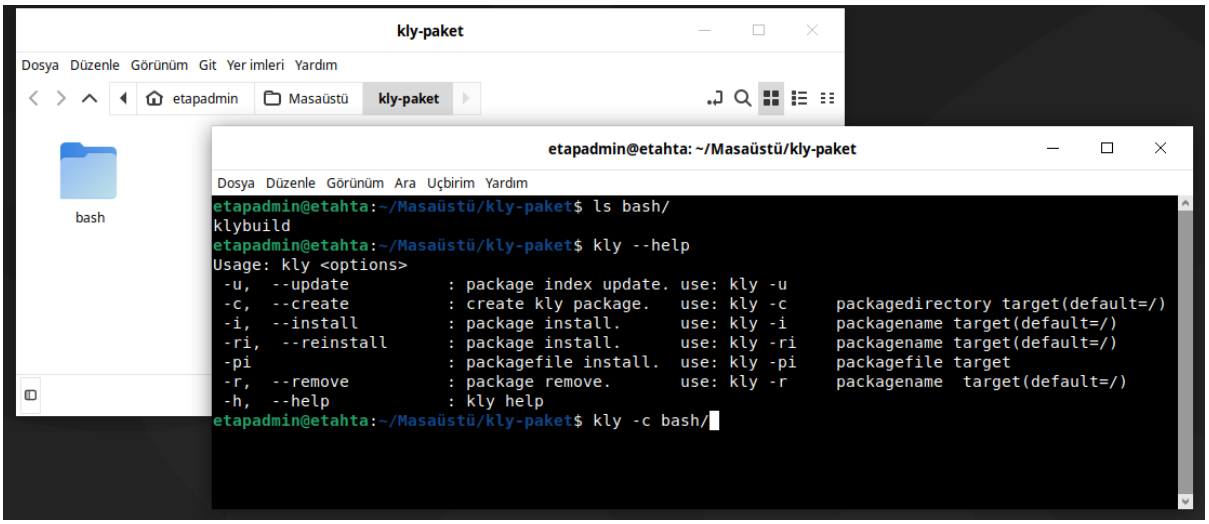
Değişkenleri dereleme scripleri içinde kullanılmaktadır. Örneğin, kaynak dizinde işlem yapmak için sadece **$SOURCEDIR** kullanmanız yeterlidir. Bu yapılar tüm paketlerde geçerli olacak.

Not: Bazı paketlerin ek dosyaları olabilir. Derleme scripti altında **Ek dosya için tıklayınız** bağlantısını(link) kullanarak ek dosyaları indirin ve paketin dizini içine çıkartınız. **bash** paketinin ek dosyaları olsaydı **bash** dizini içine indiğimiz dosyayı arşivde çıkartacaktık.

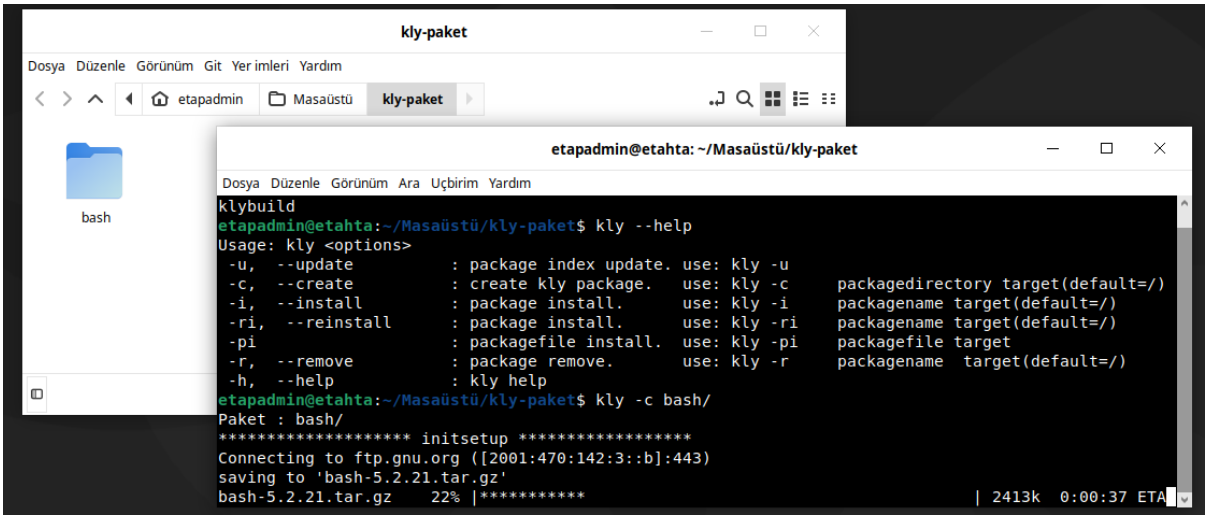
kly-paket dizini konumunda aşağıdaki gibi terminal açınız.



Açılan terminalde aşağıda görüldüğü gibi komutu çalıştırınız. komut hangi konumda çalıştırılmışsa **.kly** uzantılı pakeyimiz orada oluşacaktır. Siz istediğiniz yerde çalıştırabilirsiniz. Önemli olan paket için oluşturduğunuz **klybuild** dosyanınızın konumunu doğru vermeniz.

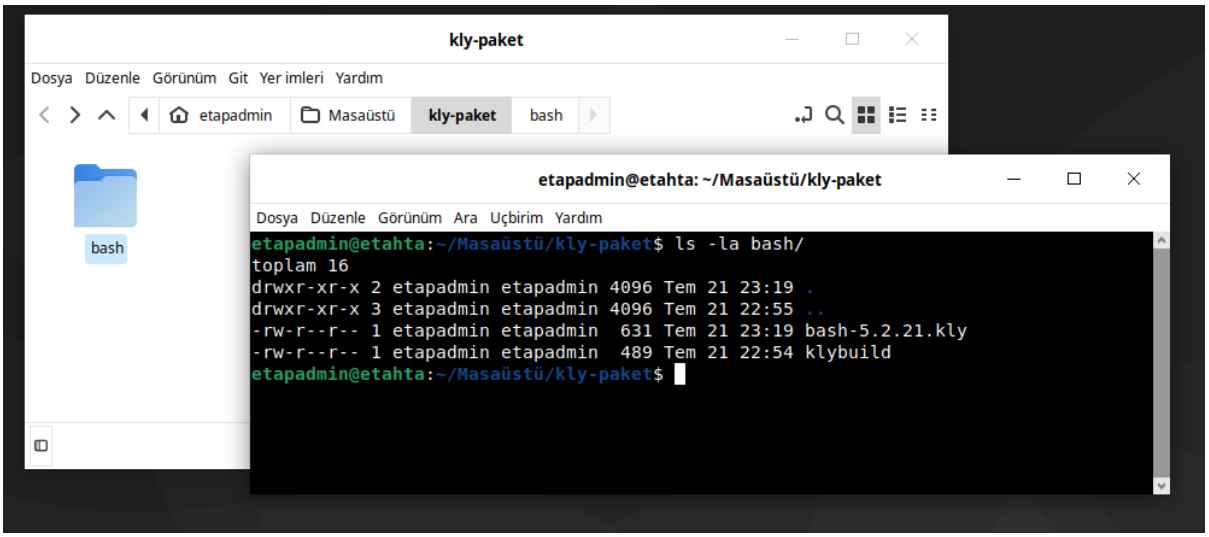


Aşağıda **bash** dizinini parametre olarak vererek **bash** paketimizin derlemesini başlatıyoruz.



Derleme işlemi paketin büyüklüğüne bağlı olarak zaman alacaktır. Paket derlemesi bittikten sonra aşağıda görüldüğü gibi termina çıktısı almalısınız. Sorun çıkması durumunda terminalde hata mesajları alırsınız.

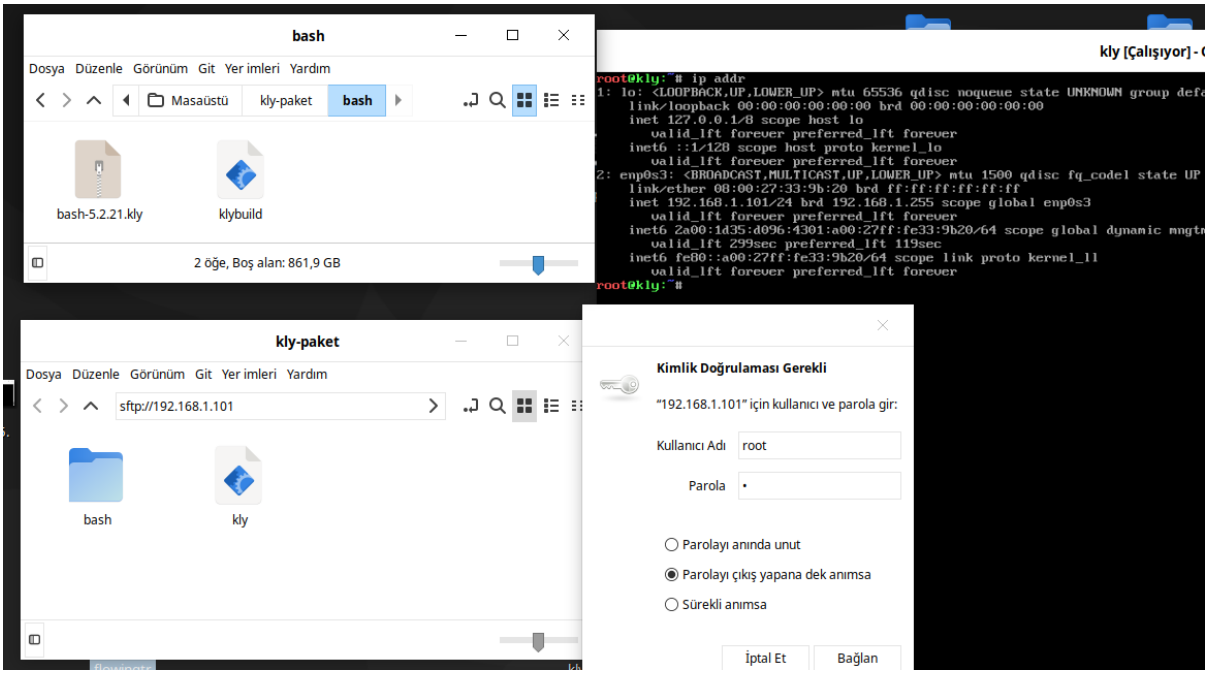
Derleme işlemi bittikten sonra **kly-paket/bash** dizini komununda aşağıda görüldüğü gibi **bash-5.2.21.kly** paketimizi oluşturacaktır.



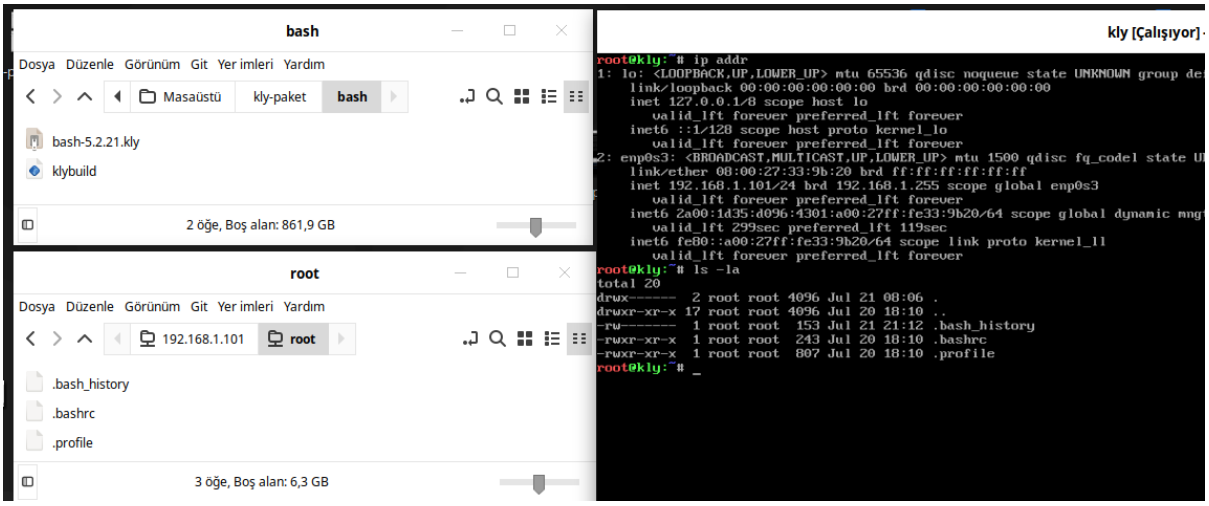
kly ile Paket Kur

kly Paket Sistemi ile paket oluşturma konusunda **bash** paketi derlenmişti. Şimdi ise bu paketi **Temel Sistem** üzerine kopyalamayı ve kurma işlemini yapalım. **bash** paketimiz masaüstümüzde **kly-paket/bash/bash-5.2.21.kly** konumunda bulunmaktadır. Bu konumdaki dosyamızı **Temel Sistem** üzerine **scp** veya **sftp** kullanarak kopyalayabiliriz. **scp** terminal üzerinden kopyalar. **sftp** terminal veya pencere yöneticisi üzerinden kopyalar. Burada tercih size kalmıştır. **sftp** ile pencere yöneticisini kullanmak daha kolay olabilir. **scp ve sftp** kullanımı **Yardımcı Konular** bölümünde detaylıca anlatılmıştır. Ayrıca **Temel Sistemin Hazırlanması** bölümünde de **scp** ve **sftp Temel Sistem** ile nasıl kullanılacağı anlatıldı.

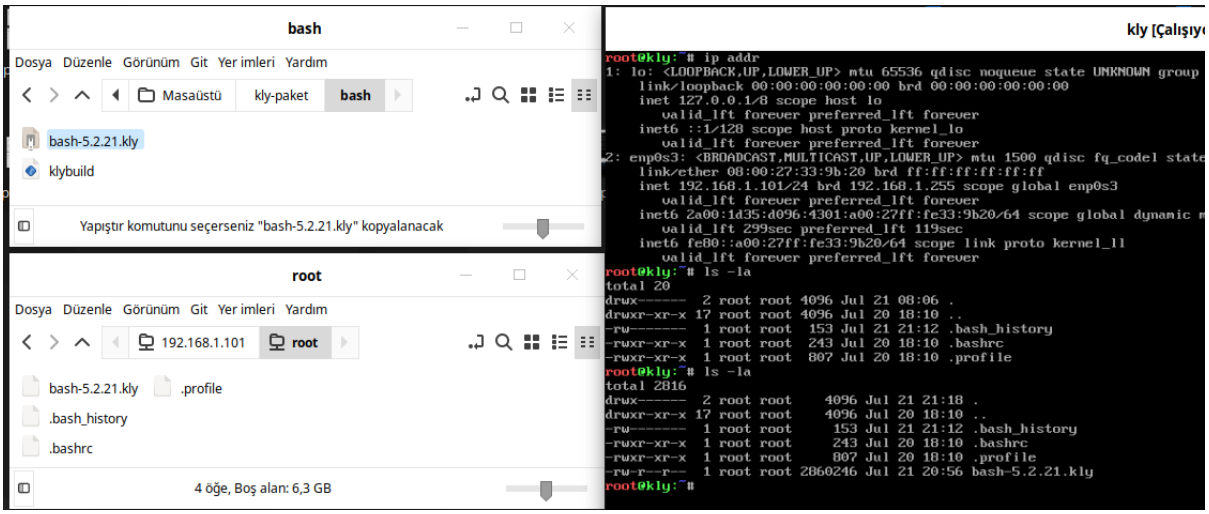
Aşağıda **sftp** ile nasıl bağlantı yapılacağı görülmektedir.



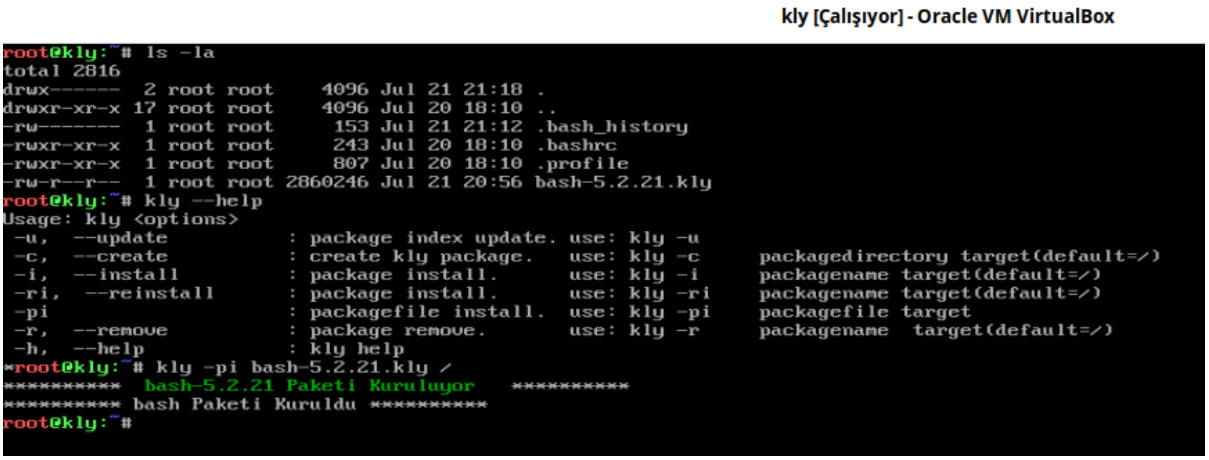
Aşağıda **sftp** ile **bash-5.2.21.kly** paketini kopyalanma öncesi **Temel Sistemde** olup olmadığını **ls -la** komutuyla kontrol ediyoruz.



Kopyalama işleminden sonra **bash-5.2.21.kly** paketinin **Temel Sistemde** olup olmadığını **ls -la** komutuyla görüyoruz.



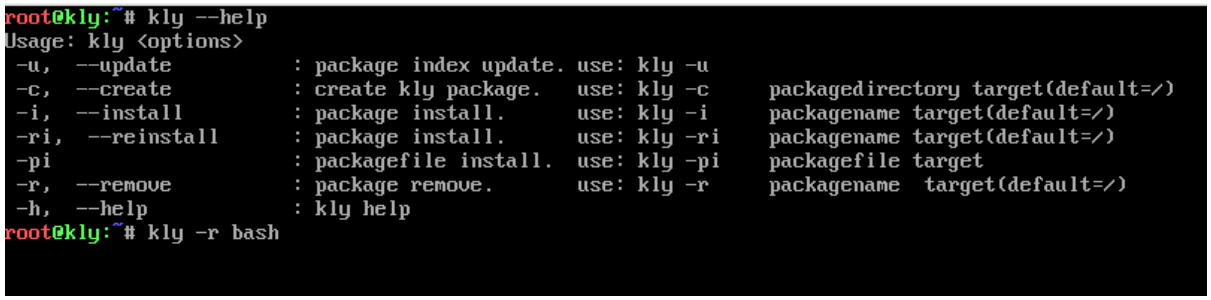
kly paket sistemini kullanarak yerelde bulunan paketin kurulumunu **kly -pi bash-5.2.21.kly /** komutuyla yapıyoruz.



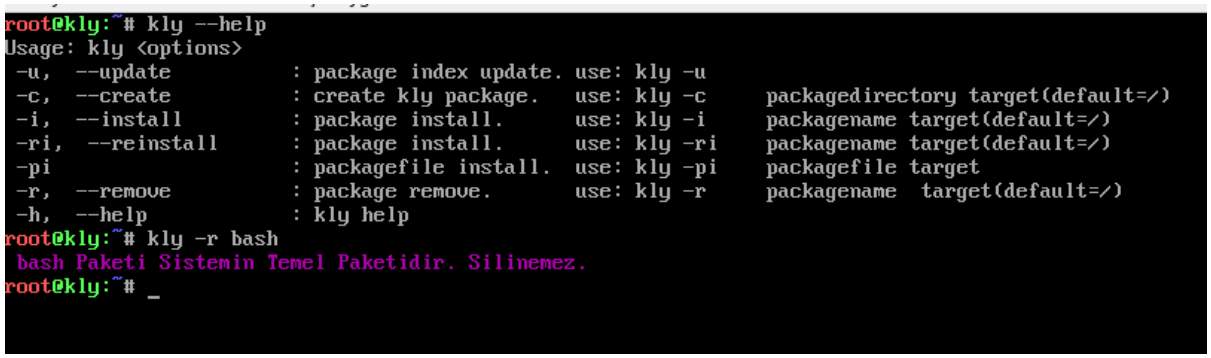
kly ile Paket Kaldir

kly Paket Sistemiyle sistemde yüklü olan bir paketi **kly -r paketadı** şeklide komut kullanılarak kaldırılabilir. Daha önce paket oluşturma ve paket kurulum işlemlerinde **bash** paketini kullanmıştık. Şimdi **bash** paketini kaldırmamız durumunda sistemde terminali kulanamaz duruma getirecektir. Bazı temel paketleri kaldırmak sistemimizin bozulmasına sebep olabilir. Paket kaldırırken dikkatli olmak gerekir.

Aşağıda **bash** paketinin nasıl kaldırılacağı görülmektedir.



Bazı paketleri **(glibc, ncurses, readline, bash)** tasarladığımız **kly** paket sisteminde kaldırılmasını engelledik. Bu paketler kalması durumunda sistem kullanılamaz hale gelir. Eğer değişiklik gerekiyorsa sadece yeniden kurulum yapılabilir. Aşağıda paketin kaldırılamadığı mesajı görülmektedir.



glibc, ncurses, readline, bash dışında başka paketler olsaydı paket kaldırılması gerçekleşecekti.

kly Paketlerini Githuba Yükleme ve İndexleme

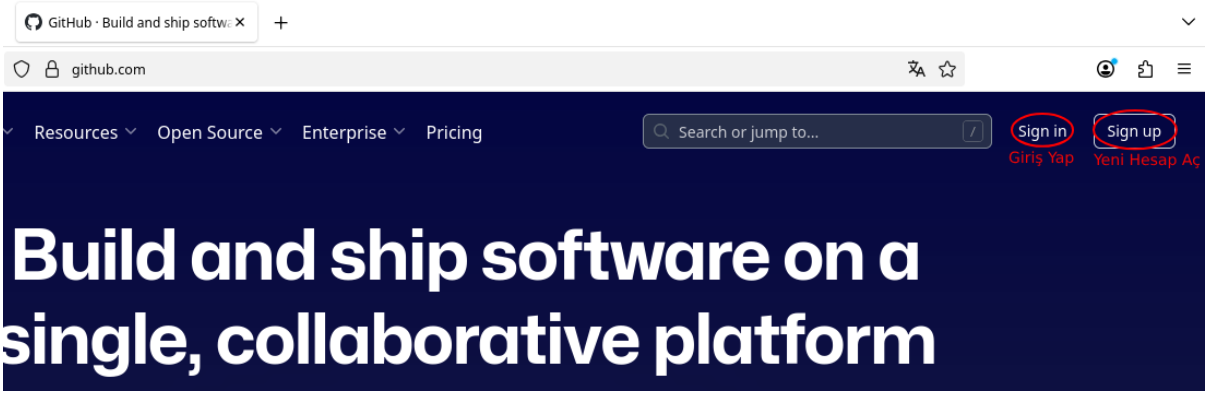
Depo, paketlerimizin olduğu alandır. Depoda ne kadar paket varsa bunların isimleri sürüm numaraları gibi bilgiler ile adreslerini liste halinde oluşturma işlemine **depo indexleme** denir. Depo indexlenirken genellikle bilgiler **paket derleme talimatı(klybuild)** dosyasından alınır. Paketlerin listesi, paketler kurulurken, silinirken ve güncellenirken kullanılmaktadır.

kly github Depo Yapma

Bu doküman kullanılarak hazırlanan paketleri bilgisayarınızda bir dizinde tutabiliriz. Fakat bu çok kısıtlı bir sistem olmasına sebep olacaktır. Paketleri bir internet ortamında bir yerde saklayarak, kurmak istediğimizde internet(uzak) üzerinden kurulması daha doğru bir yöntemdir. Bu dağıtımda paketlerimizi github.com üzerinde oluşturulan bir repository üzerinden çekilmektedir. İnternetteki paketlerimizin listesi her yeni paketi yükleme sırasında güncellenmektedir. Bu işlem github hesabı üzerinden yapılmaktadır. github hakkında temel işlemler için github konusunu okuyunuz.

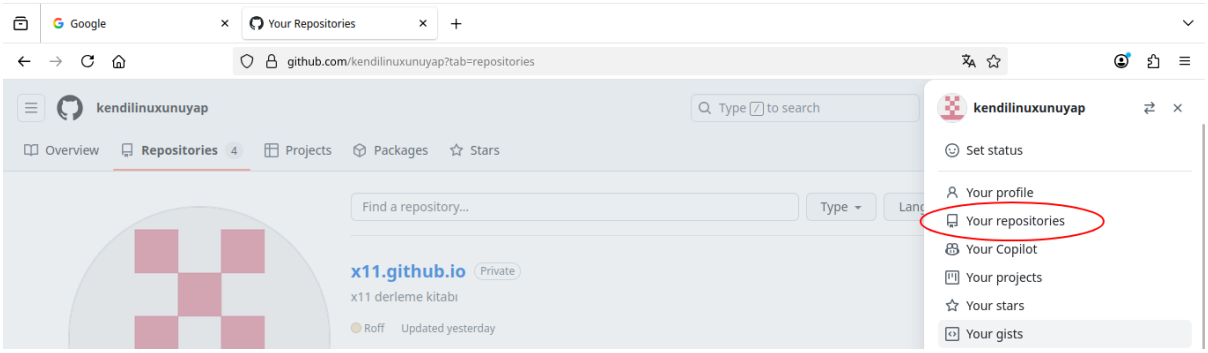
github Sitesi Açılır

Singup seçeneği seçilerek bir hesap oluşturulur. Biz bu dokumanda kullandığımız kendilinuxunuyap hesabını açtık.



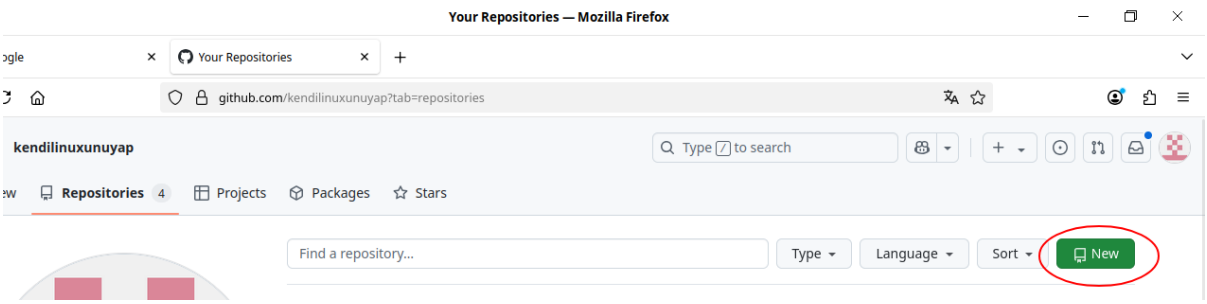
github Üzerinden Hesaba Giriş Yapılır

github hesabı açılır(kendilinuxunuyap) ve sağ tarafta bulunan menüden **Your repostrores** seçilir.

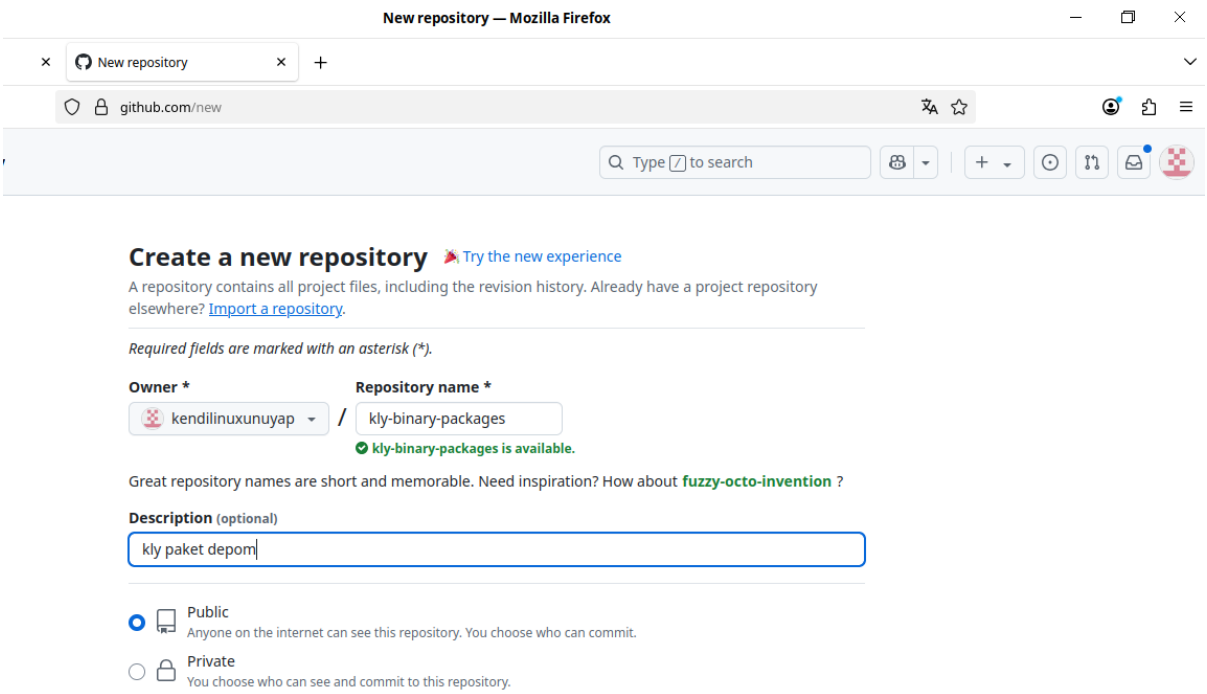


Yeni Depo Alanı Oluşturulur

Karşımıza gelen ekrandan **New** Seçeneği seçilir.

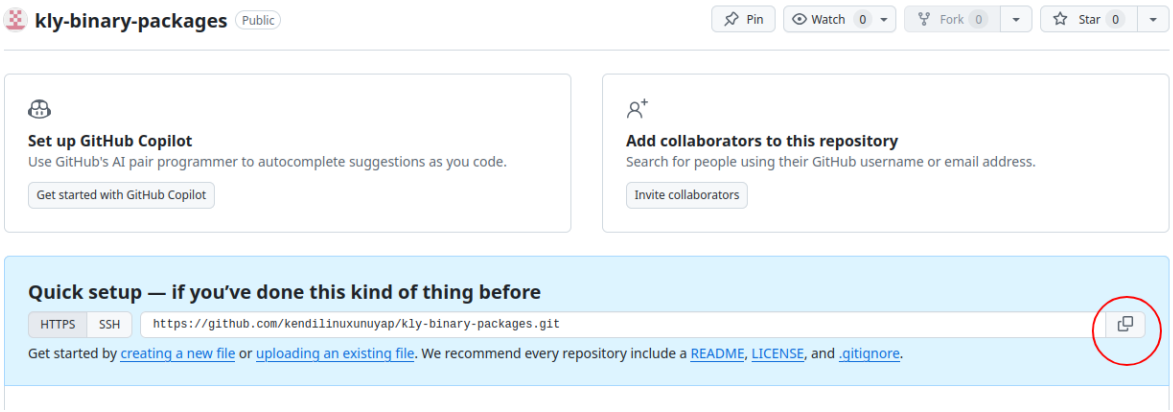


github repository oluşturulur(kly-binary-packages)



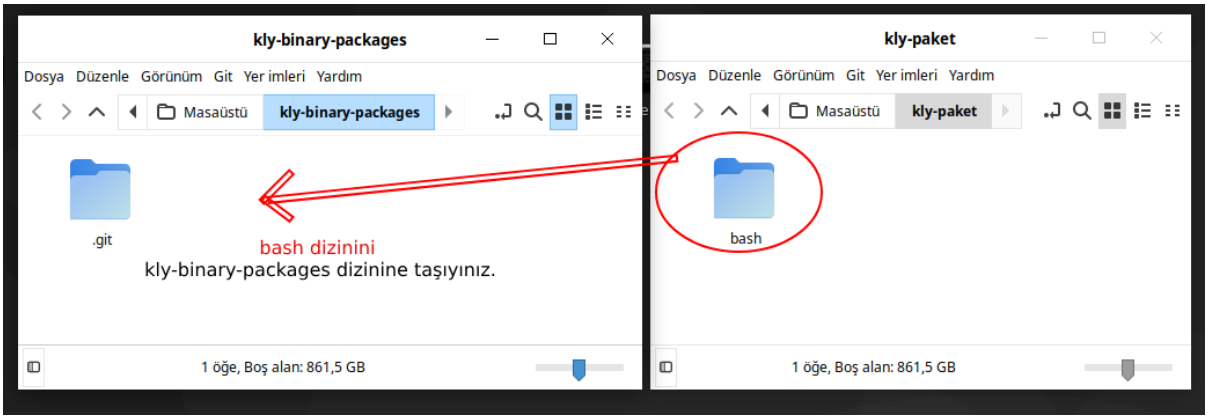
kly-binary-packages Depomuz Yerele(Bilgisayara) Kopyalanır(Clone)

kly-binary-packages adresi kopayanır.

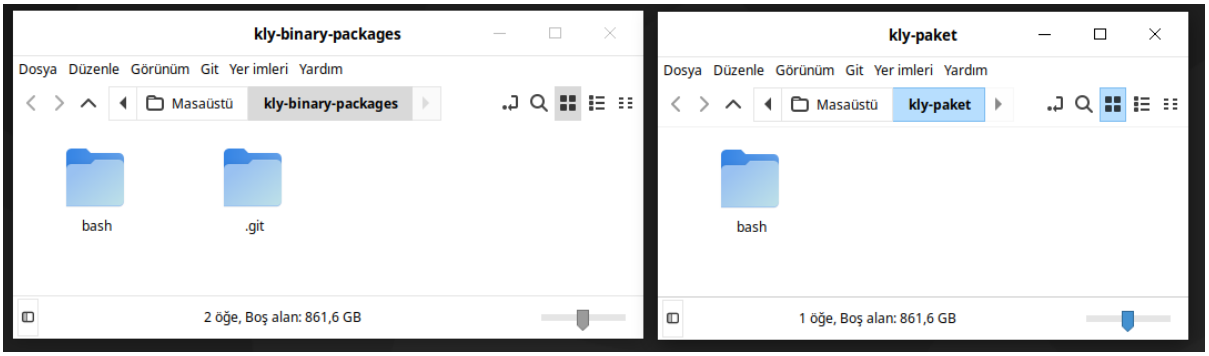


Yerelde(bilgisayarda) istediğiniz yere(masaüstünü tercih ettim) indirilir(klonlanır/download).

Paketimiz kly-binary-packages Dizinine Kopyalanır

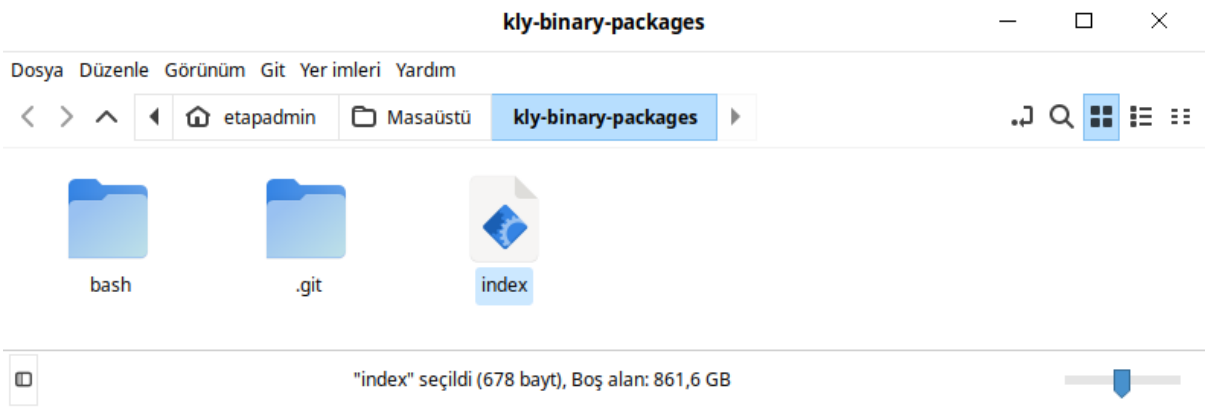


bash paketi aşağıdaki gibi taşınır.



index Dosyası Oluşturulur

Aşağıdaki script kly paket dosyalarımızı olduğu dizinde tek tek açarak içerisinden **klybuild** dosyalarını çıkartır. Paketle ilgili bilgileri alıp **index.lst** dosyası oluşturmaktadır. İstersek paketler local ortamdada index oluşturabiliriz. Bu dokümanda github üzerinde oluşturacak şekilde anlatılmıştır. Paket indeksi oluşturan **index.lst** dosyası aşağıdaki gibi olacaktır. Listede name, version ve depends(bağımlı olduğu paketler) bilgileri bulunmaktadır. Bilgilerin arasında **|** karekteri kullanılmıştır.

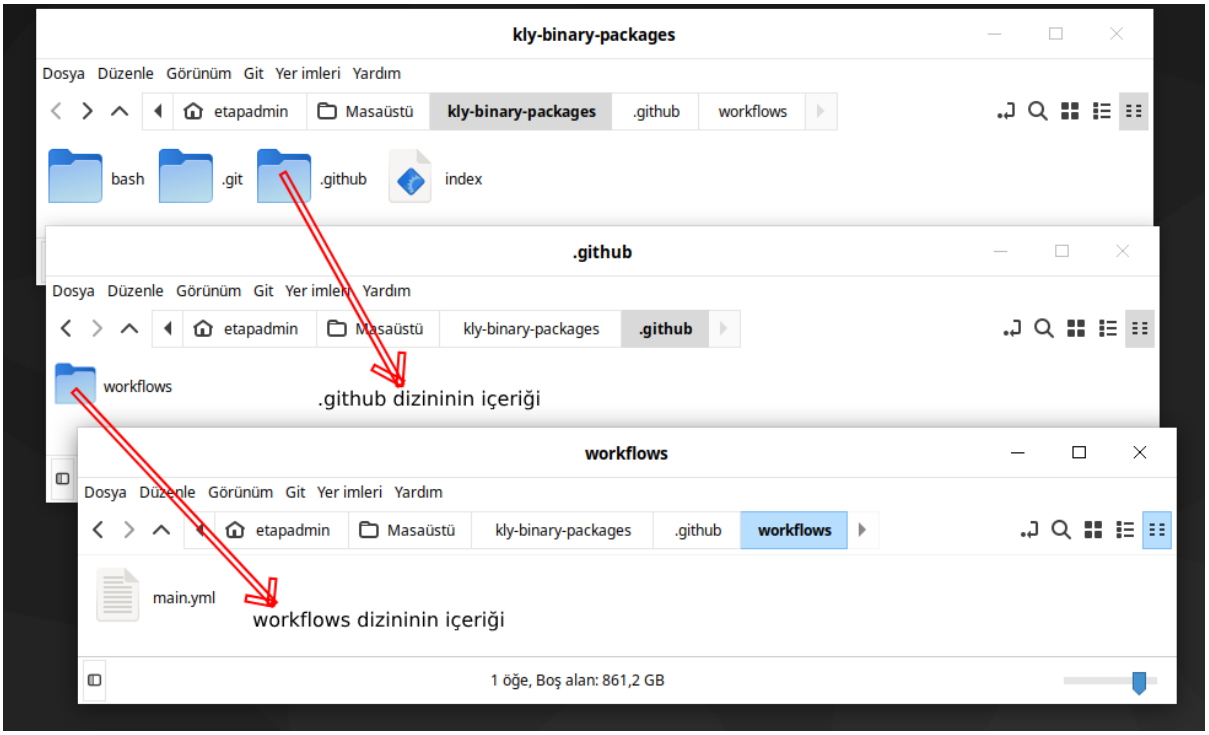


Yukarıda gösterildiği gibi **kly-binary-packages** dizininde aşağıda verilen **index** dosyasını oluşturunuz. Aşağıdaki script kodlarını **index** dosyasının içerisine ekleyin.

```
#-----------------------------------------------------------------------------------------------------------------------------
#!/bin/sh
#set -ex
mkdir /output -p
mkdir -p /klysource
>index.lst
find * -type f -name *.kly |
                while IFS= read file_name; do
                        dosya="$(dirname $file_name)/klybuild"
                        version=$(cat $dosya|grep version=)
                        name=$(cat $dosya|grep name=)
                        depends=$(cat $dosya|grep depends=)
                        echo "$name|$version|$depends|$(dirname $file_name)">>index.lst
                done
cp -rf index.lst /output

# ***************************source files*****************************
cp -prfv ./* /klysource/

find /klysource/* -type f -name *.kly |
                while IFS= read file_name; do
                rm -rf "$file_name"
                done
tar -cf /output/klysourcepackage.tar /klysource/
rm -rf /klysource
```

main.yml Dosyası Oluşturulur

github'a dosya gönderdiğimizde **index** bash scriptimizi çalıştırması için aşağıda gösterilen şekilde **kly-binary-packages** dizinine **.github/workflows** dizinini oluşturun. **.github/workflows** dizini içine **main.yml** dosyasını oluşturunuz.



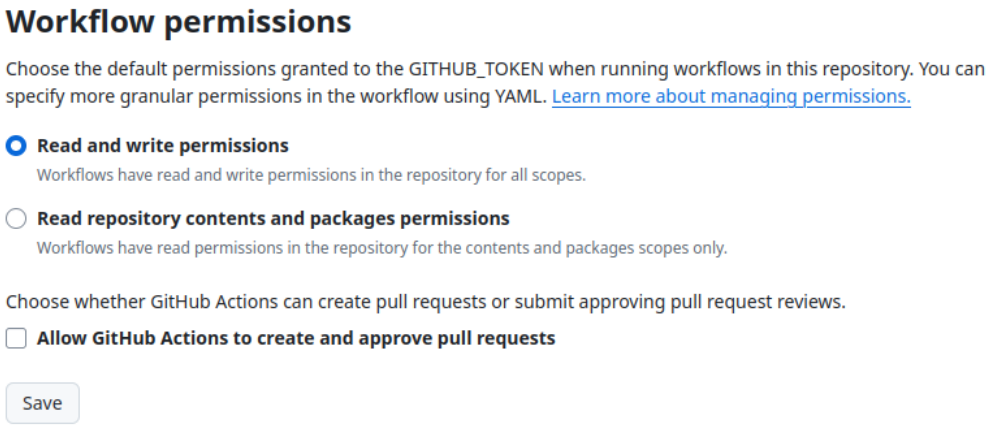
main.yml dosyasısdaki **sh index** satırı **index** scriptimizi her githuba paket gönderdiğimizde(commit) çalışacak ve **index.lst** dosyasını oluşturacaktır. **main.yml** içeriğine aşağıdaki kodları ekleyiniz.

```
#-------------------------------------------------------------------------------------------------------------------------------
name: CI

on:
  push:
        branches: [ master ]
  schedule:
        - cron: "0 0 1 2 6"

jobs:
        compile:
            name: depoindex
            runs-on: ubuntu-latest
            steps:
              - name: Check out the repo
                uses: actions/checkout@v2
              - name: Run the build process with Docker
                uses: addnab/docker-run-action@v3
                with:
                    image: debian:testing
                    options: -v ${{ github.workspace }}:/root -v /output:/output
                    run: |
                        cd /root
                        sh index
              - uses: "marvinpinto/action-automatic-releases@latest"
                with:
                    repo_token: "${{ secrets.GITHUB_TOKEN }}"
                    automatic_release_tag: "current"
                    prerelease: false
                    title: "Latest release"
                    files: |
                      /output/*
```

Not: Burada **main.yml** dosyasında **[master]** ifadesi **master** dalında çılışıldığını ifade eder. Eğer faklı dalla açılışıyorsak buradaki **[master]** yerine kullandığınız dalı yazınız.

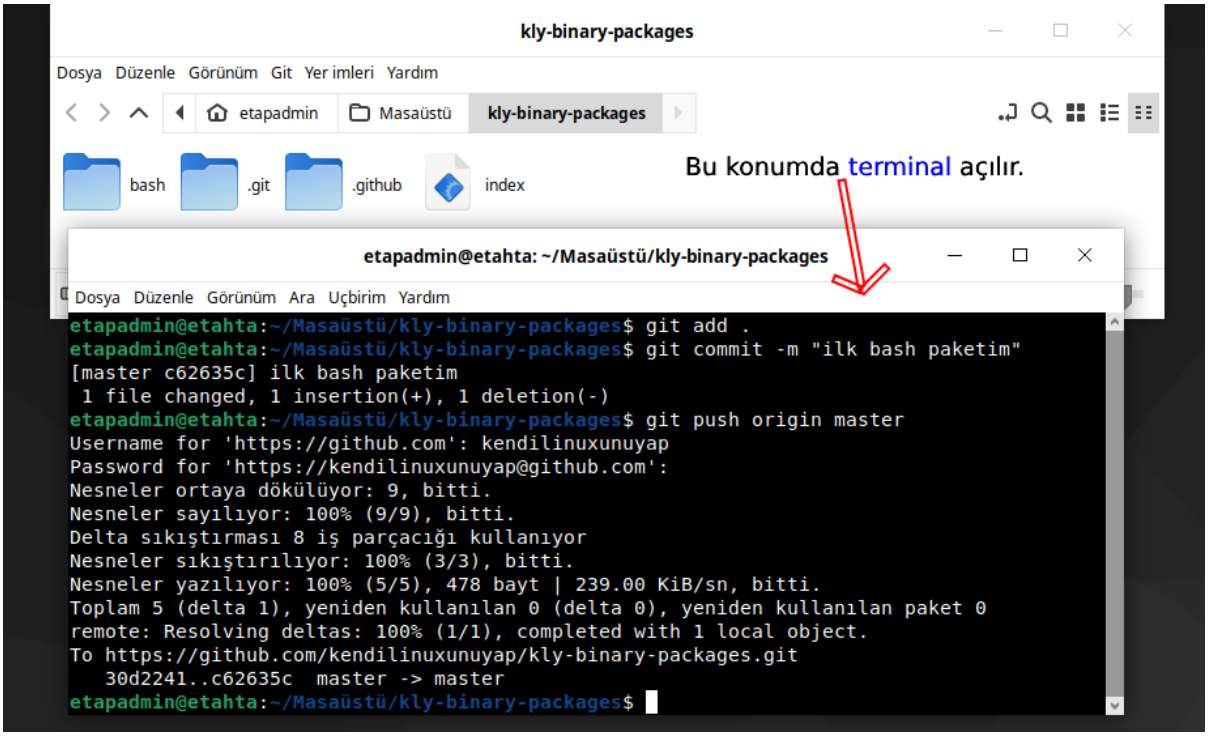
github Dosya oluşturma İzni Verin

İnternet üzerinden **kly-binary-packages** reposunda settings->action->general->Workflow permissions->Read and write permissions işaretlenmelidir.



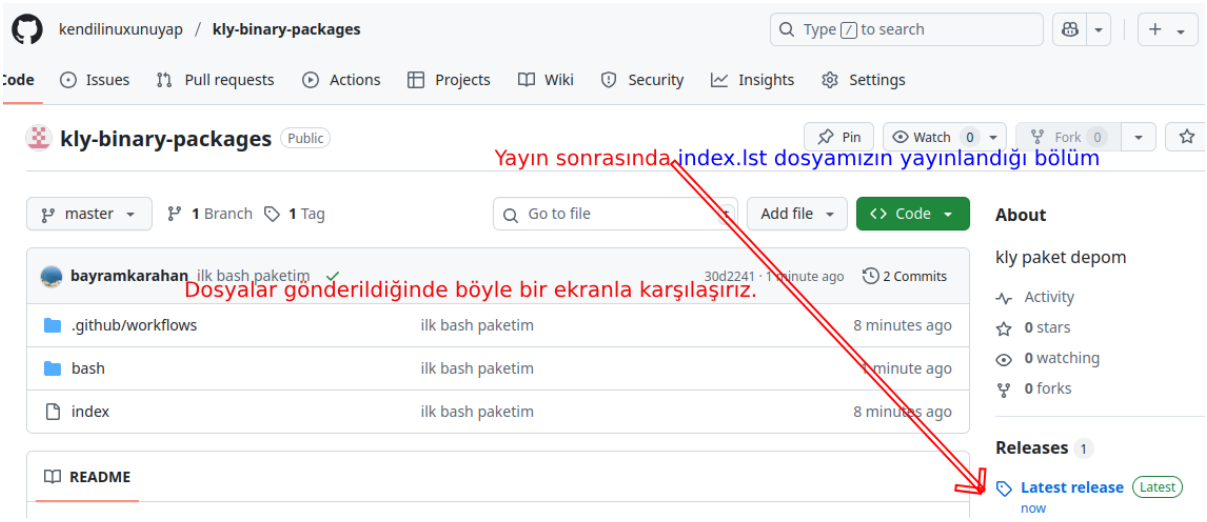
github'a kly-binary-packages Dizinini Yükleyelim(Uplod/Commit)

Yerelde **kly-binary-packages** dizini içeriğini github üzerine aşağıdakı gibi gönderilir.



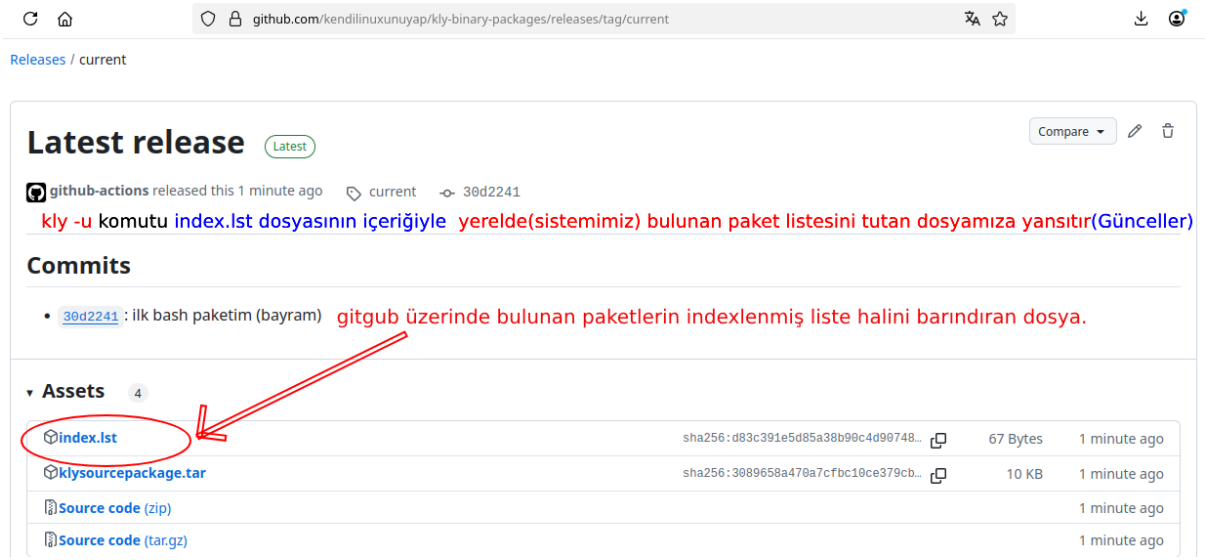
github kly-binary-packages Depo Kontrolü

Depomuza gönderdikten sonra aşağıdaki gibi gözükecektir.



index.lst İçeriği

https://github.com/kendilinuxunuyap/kly-binary-packages/releases/download/current/index.lst adresinde bulunan dosya aşağıdaki gibi liste oluşturacaktır.



index.lst içeriği aşağıdaki gibidir. Tek paket olduğu için(sadece bash) bu şekilde gözüküyor.

```
name="bash"|version="5.2.21"|depends="glibc,readline,ncurses"|bash
```

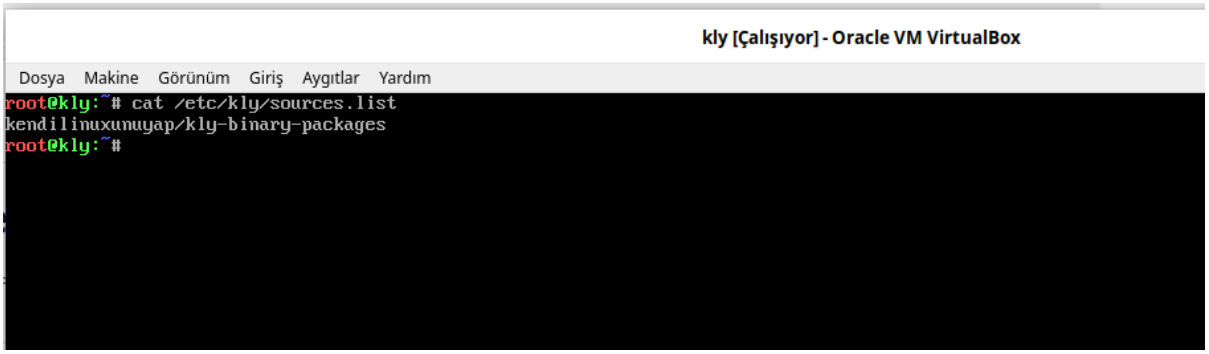
Birden fazla paketin olması durumunda aşağıdaki gibi gözükecektir.

```
name="acl"|version="2.3.1"|depends="attr"|acl
name="attr"|version="2.5.1"|depends=""|attr
name="audit"|version='3.1.1'|depends=""|audit
name="bash"|version="5.2.21"|depends="glibc,readline,ncurses"|bash
```

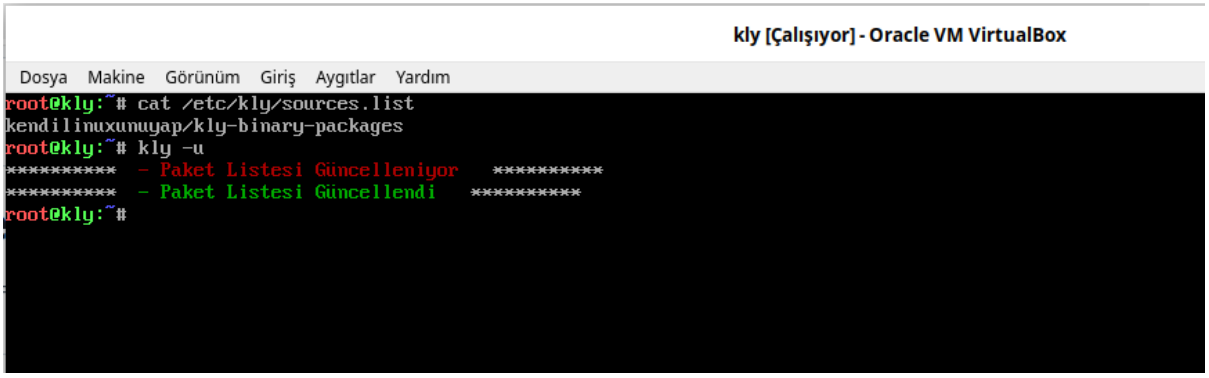
kly ile Paket Listelerini Güncelleme

kly Paket Sistemi yerelde **.kly** paketlerini **kly -pi paket.kly** şeklinde kurulabilir. Ama bütün paketlerin yerelde olması bu sistemi kullanacak kişi sayısını sınırlandıracak ve birkaç kişiyi geçmeyecektir. Paketleri internet ortamında tutmak sistemi kullanacak kişilerin sayısnı artıracak ve istenilen zaman ve konumda paketlere erişim imkanı sunacaktır.

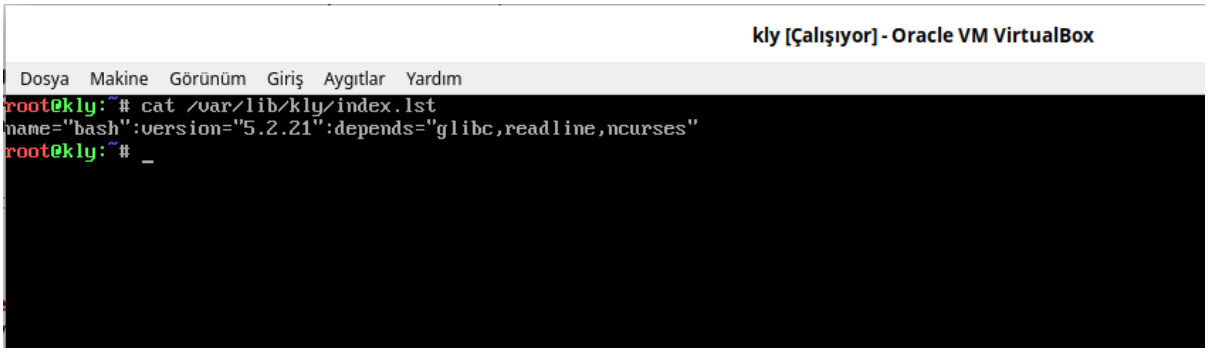
kly paketleri github üzerinde tutulmaktadır. Bu paketlerin listesini tutan index dosyası github ortamında **https://github.com/kendilinuxunuyap/kly-binary-packages/releases/download/current/index.lst** adresinde tutulmaktadır. Bu adresi sisteme kaydetmeliyizki güncelleme sırasında bu adreslerden güncel **index.lst** dosyasını yerele indirebilmeli. Adreslerin listesini tutan dosyamız **/etc/kly/sources.list** konumunda turulmaktadır. Debianda da benzer(/etc/apt/sources.list) durum bulunmaktadır.



Yerelde **(Temel Sistem)** ise **/var/lib/kly/index.lst** dosyamızda paketlerin listesi tutulur. Mevcut sisteme yeni bir paket yaptık ve github'a yüklediğimizi varsayalım. Bu durumda yerelde **(Temel Sistem)** ise **/var/lib/kly/index.lst** konumundaki dosyanında https://github.com/kendilinuxunuyap/kly-binary-packages/releases/download/current/index.lst adresindeki dosyayla aynı olması gerekir. Bu senaryo tüm linux dağıtımlarında aynıdır. Bu sebeplerden dolayı bir paket kurulum öncesi genelde **güncelleme** işlemi yapılır. Aşağıda güncelleme işleminin yapıldığını göstermektedir.



Aşağıda **/var/lib/kly/index.lst** konumundaki dosyanın içeriğini görmekteyiz.



apt Paket Sistemi

apt paket sistemi debian tabanlı sistemlerde paket yönetimi için kullanılan bir uygulamadır. Temel işlemler şunlardır.

Yerel Paket İndexini Güncelleme

```
sudo apt  update
```

Paket Yükleme

```
sudo apt install paket_adi
```

Paket Silme

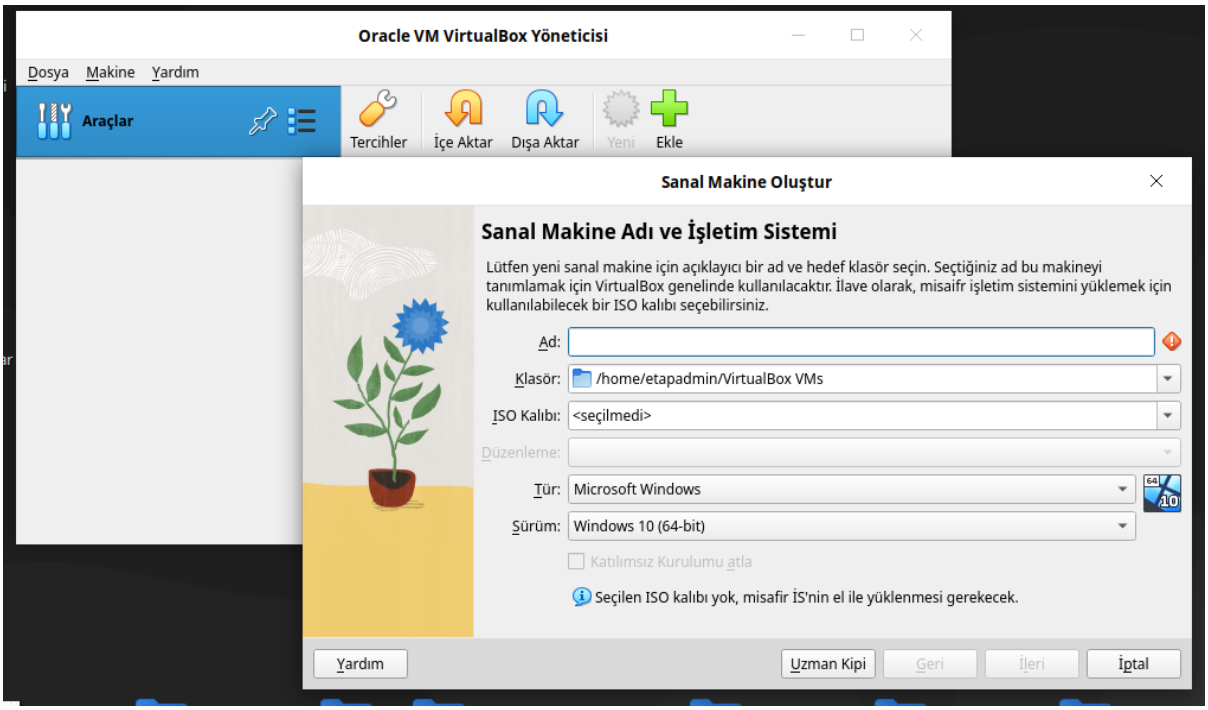
```
sudo apt remove paket_adi
```

VirtualBox

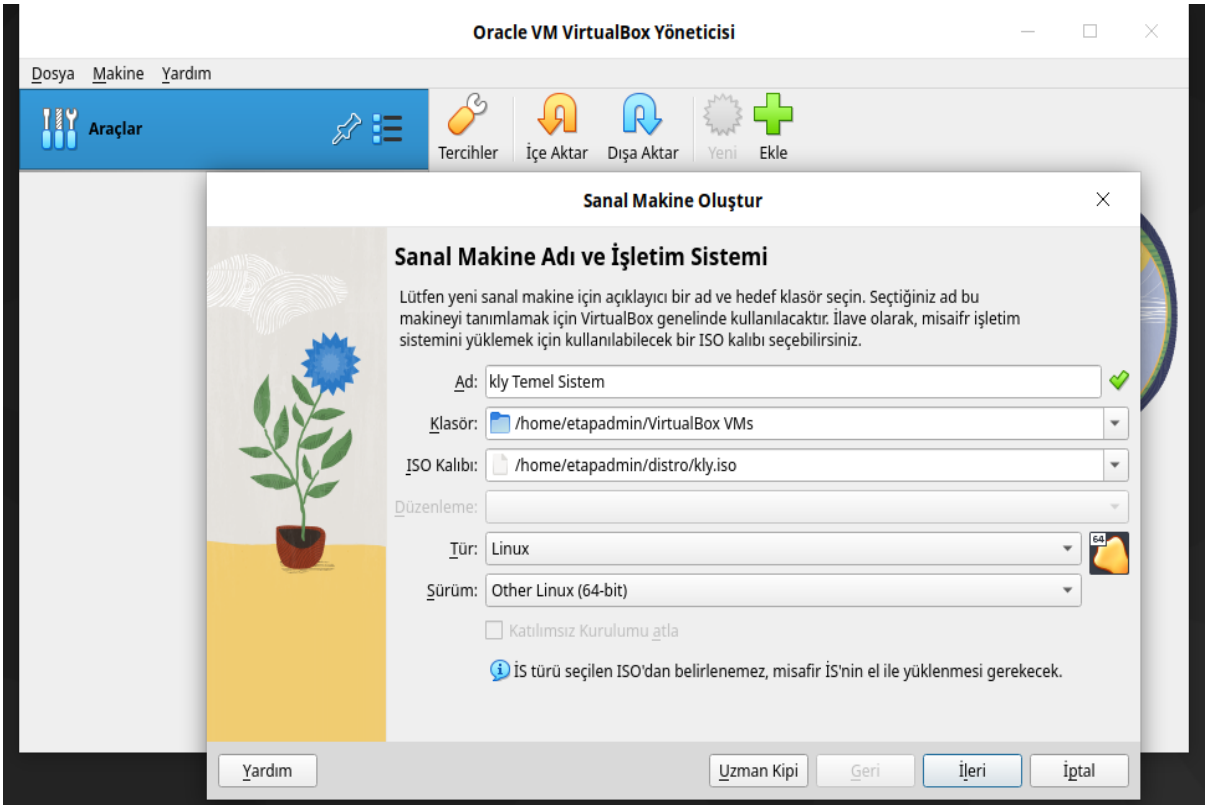
VirtualBox, Oracle tarafından geliştirilen bir açık kaynaklı sanallaştırma yazılımıdır. Bilgisayarınızda çalışan bir işletim sisteminin (örneğin Windows, Linux, macOS) içinde başka bir işletim sistemi çalıştırmanıza izin verir. Bu çalıştırdığınız ikinci işletim sistemi (konuk/guest) kendi sanal donanımına sahipmiş gibi davranır.

```
sudo apt update                          # index güncellenir
sudo apt install virtualbox-7.0          # Sisteme virtualbox kurulur
```

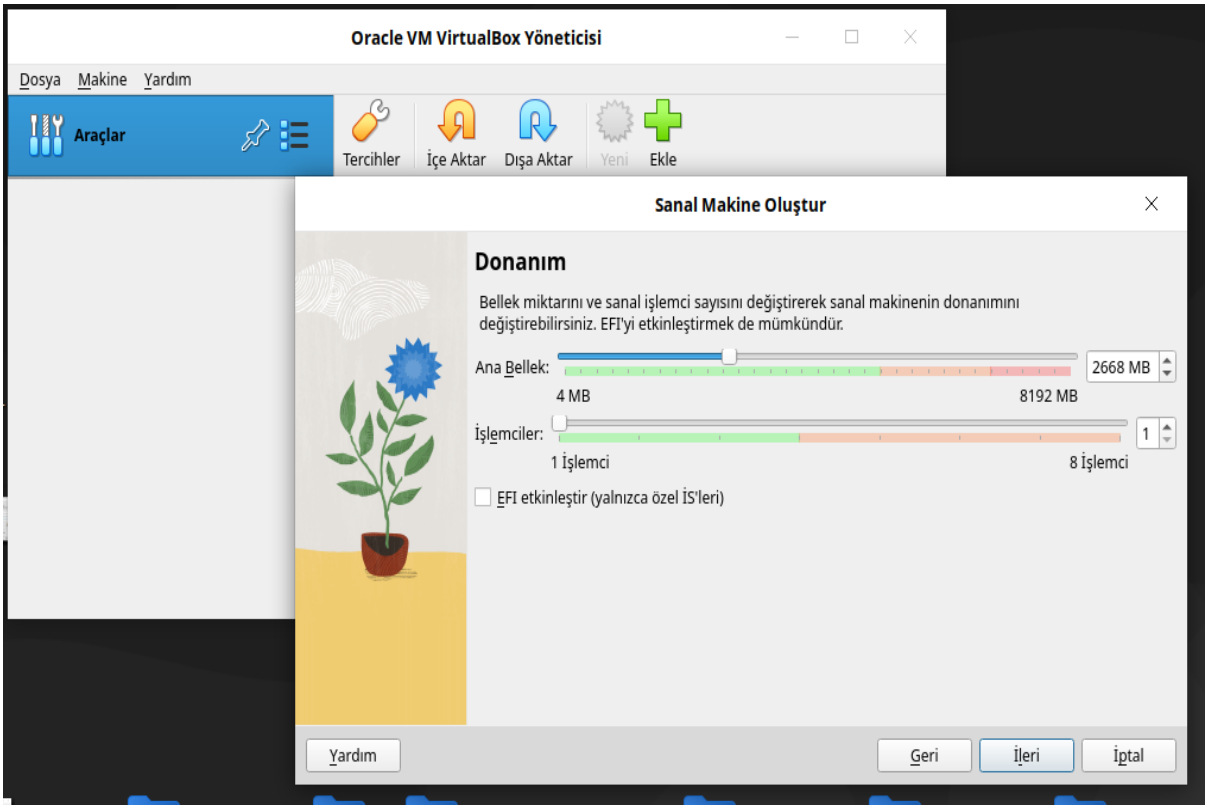
Ekle seçeneğini seçin. Aşağıdaki gibi bir ekran gelecektir.



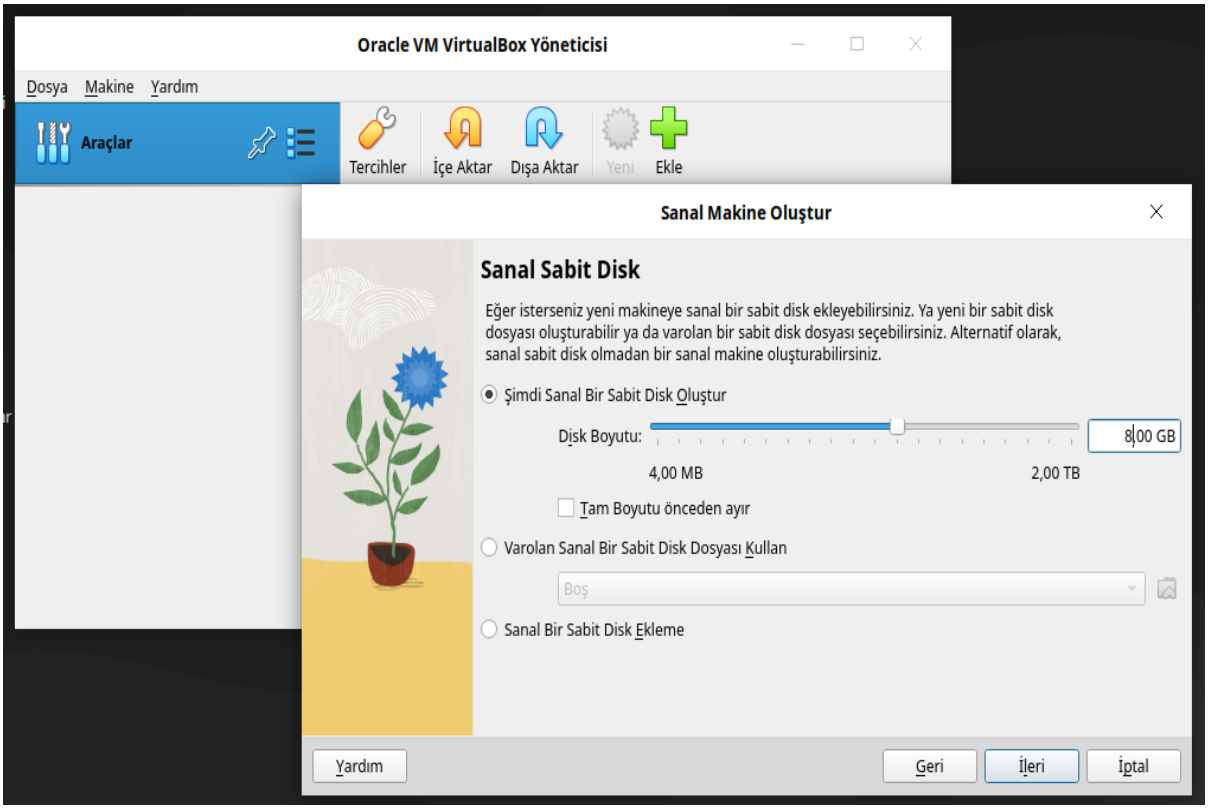
Aşağıdaki gibi seçenekleri doldurunuz.



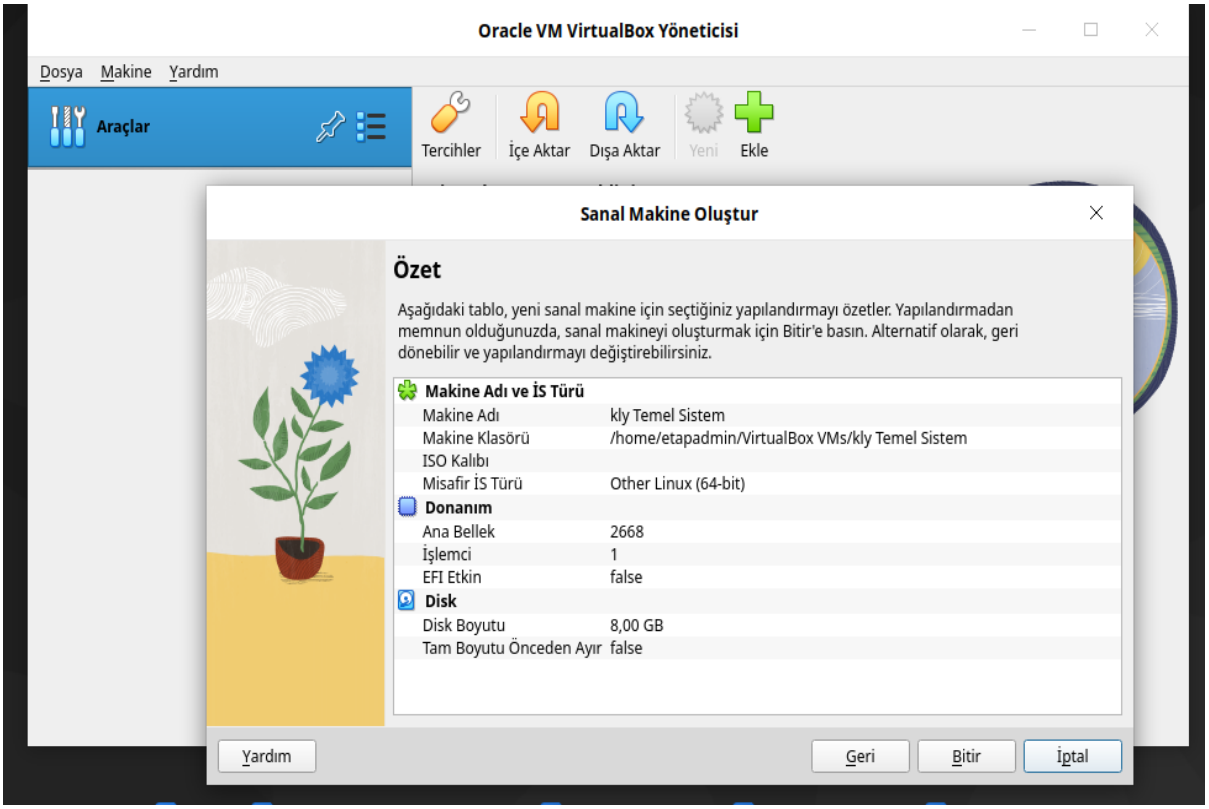
İleri seçeneği seçilir ve aşağıdaki gibi ekrana gelirsiniz. Bellek miktarını aşağıdaki gibi belirleyiniz. **İleri** seçeneği seçilir.



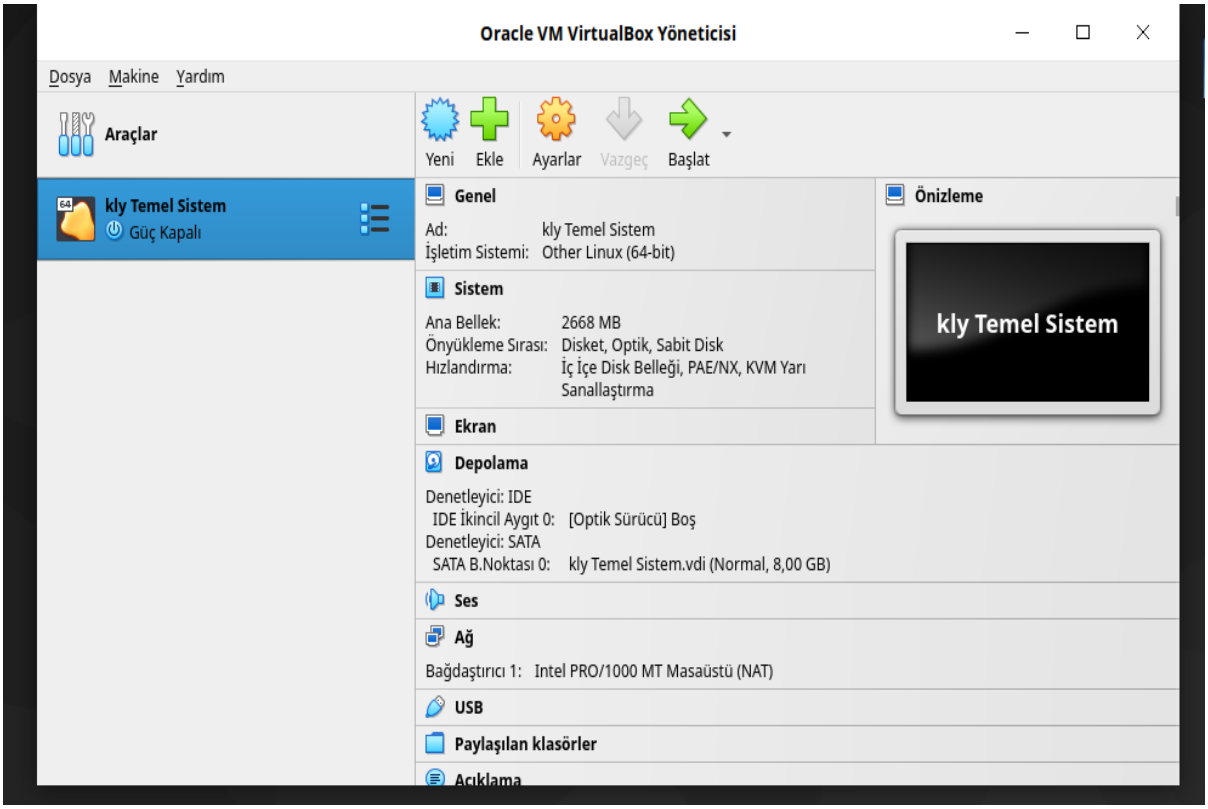
Disk boyutu belirlenir. 8GB bu sistem için yeterli. **İleri** seçeneği seçilir.



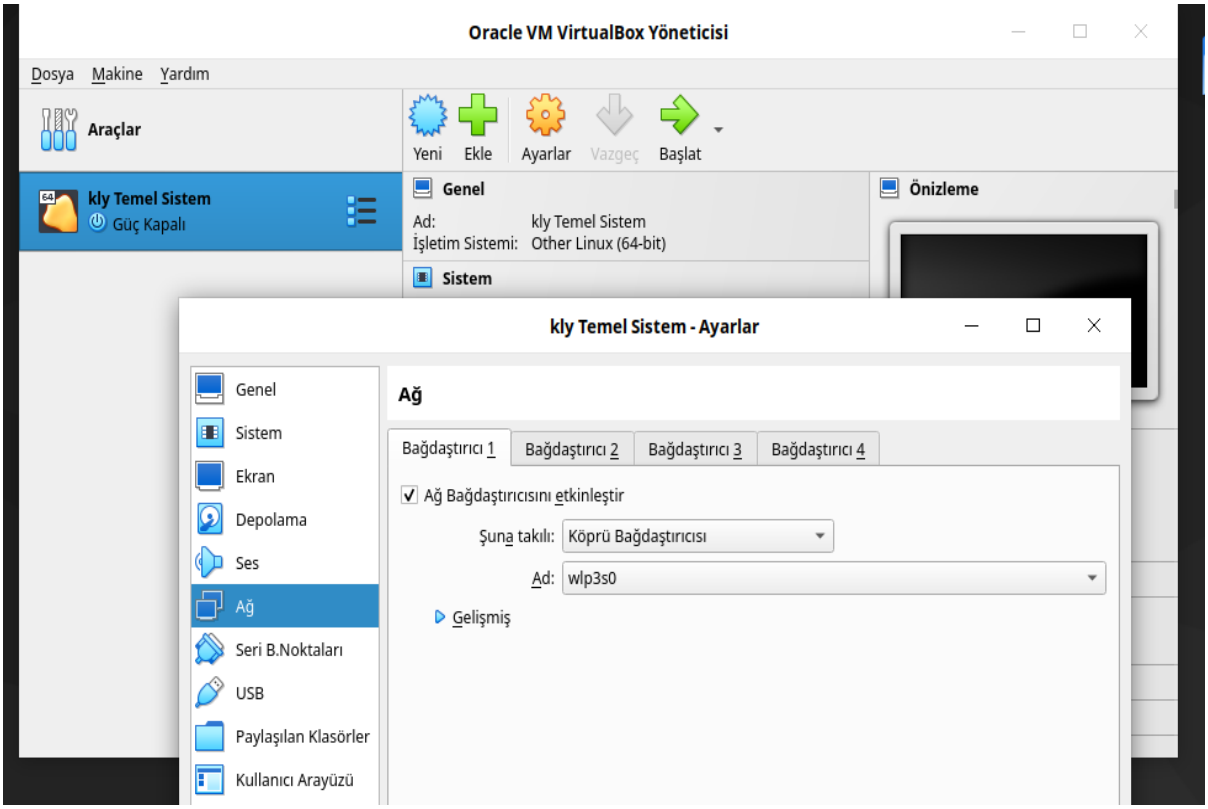
Tüm işlemler bitince aşağıdaki gibi pencere gelir. Bitir'i seçerek işlem tamamlanır.



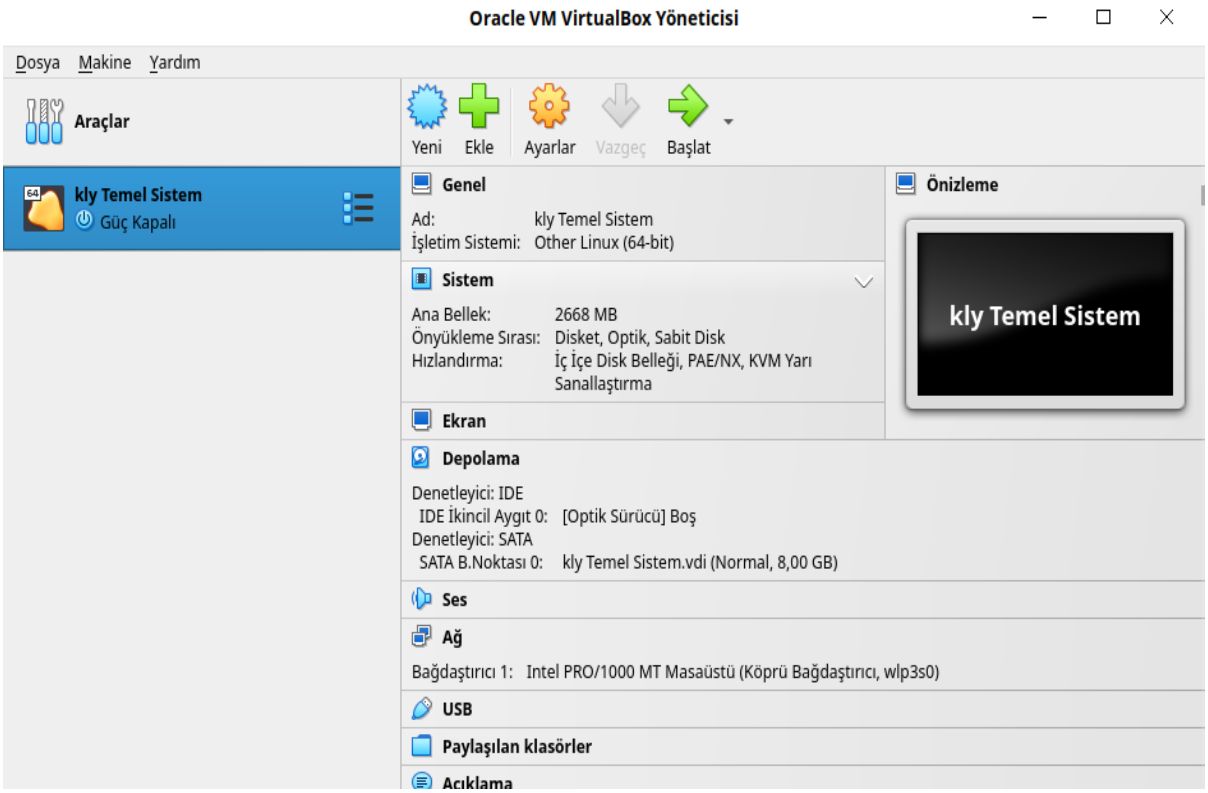
Bitir işlemi seçildikten sonra aşağıdaki gibi görünecektir.



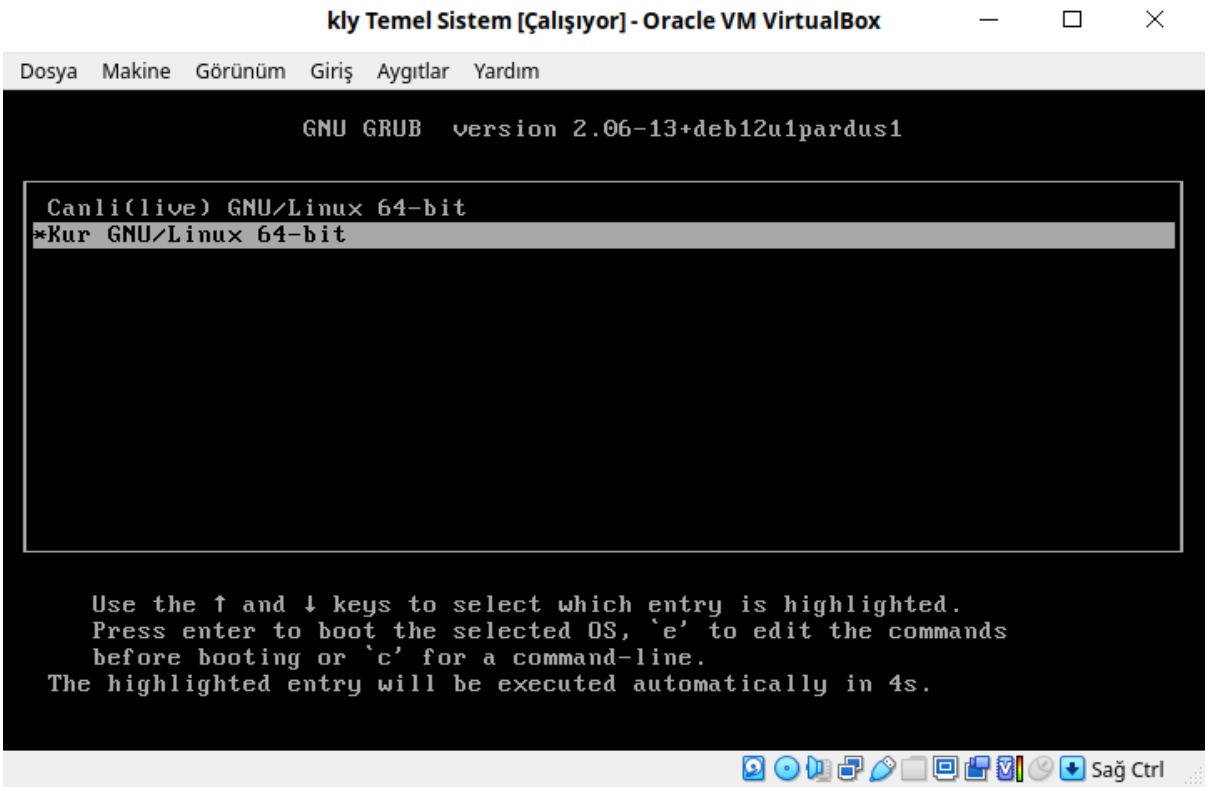
Ayarlar bölümünde aşağıdaki gibi **Ağ** ayarında değişiklik yapılır.



Başlat seçilerek sistem çalıştırılır.



İso açıldığıda aşağıdaki gibi ekran gelecektir. Burada seçili olan iso **Temel Sistem** isosudur.



Kaynak Kod Derleme

Bir uygulamanın kodları genellikle çalışmaz(python benzeri kodlar istisna). Bu kodlardan sistemlerin çalışması için çalışabilir dosyalar üretilir(linuxta ikili dosya, elf, windowsta exe, com vb.). Bu çalışabilir dosyaları koddan oluştururken iki faklı şekilde oluşturabiliriz.

- **Paylaşımlı Derleme(dynamic):** Kendine lazım olan kütüphaneleri sistem üzerindeki başka uygulamalarla ortak kullanır.
- **Paylaşımsız, gömülü(static):** Kendisine lazım olan kütütphaneleri kendi içinde barındırır(porable uygulama gibi).

Şimdi aşağıdaki kaynak kodumuzu iki farklı yöntemle derleyelim.

```
//main.c dosyamız
#include <stdio.h>
void main(){
    printf("Merhaba\n");
}
```

1-Paylaşımlı Derleme(dynamic):

Derlenen uygulama sistemde bulunan kütüphaneleri kullanacak şeklide derlenmesidir. Uygulama boyutu küçüktür, taşınabirliği sınırlanabilir. Aşağıdaki gibi derlenir.

```
gcc -o main main.c
```

main.c kodumuzu **main** adında ikili çalışabilir dosyaya dönüşütürdük. **ldd** komutuyla **main** dosyasının kullandığı kütüphaneler öğrenilir.

```
unset LD_PRELOAD
ldd ./main
        linux-vdso.so.1 (0x00007ffdb3bb9000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f2c53fe0000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f2c541e7000)
```

Burada **libc.so.6** ve **ld-linux-x86_64.so.2** dosyaları **glibc** tarafından sağlanır. Derlenmiş dosyanın çalışması için tüm bağımlılıklarının sistemde bulunması gereklidir.Sadece gerekli olan kütüphaneleri görmek için **readelf -d** komutu kullanılabilir. Aşağıda gerekli olan kütüphaneler listeleniyor.

```
readelf -d ./main | grep -i needed
       0x0000000000000001 (NEEDED)             Paylaşımlı kitaplık: [libc.so.6]
```

ldconfig

Sistemdeki kütüphanelerin konumlarını **/etc/ld.so.conf** dosyasına bakarak belirler.

```
#/etc/ld.so.conf dosyası
/usr/lib64
/usr/lib
/lib64
/lib
```

Kütüphanelerde değişiklik yapılmışsa ve hemen bu değişikliği sistemin görmesini istersek **ldconfig** komutu kullanılmalıdır.

2-Paylaşımsız, gömülü(static):

Derlenen uygulama sistemde bulunan ve çalışması için gerekli olan kütüphaneleri uygulama içine dahil eden bir derleme yöntemidir. Uygulamamızı static derlemek için **-static** parametresi ekleyerek derlenir.

```
gcc -o main main.c -static
```

Paylaşımlı(dynamic) derleme işleminde bağımlı olduğu dosyaları **ldd** komutunu kullanarak öğrenmiştik. Şimdi paylaşımsız derlediğimiz **main** dosyasında bağımğımlı olduğu kütüphaneleri kontrol ediyoruz.

```
ldd main
    not a dynamic executable
```

Bağımlı kütüphaneler yerine **not a dynamic executable** mesajı gördük. Bunun anlamı çalışması için hiçbir kütüphaneye ihtiyaç duymaz. Bu bir avantaj ve taşınabirliği artırır. Devevantajı ise boyutu büyük olur. İhtiyaca göre paylaşımlı veya paylaşımsız derleme tercih edilir.

Derleme Araçları

Linux sistemlerinde kodlarımızı derlemek kullanılan uygulamalara derleme araçları denir. Birden fazla araç olsada en temel derleme araçları **make**, **cmake**, **meson**, **python**.

1-make

make, yazılım geliştirme süreçlerinde sıkça kullanılan bir araçtır. Özellikle C ve C++ gibi dillerde projelerin derlenmesi ve yönetilmesi için kullanılır.

Aşağıdaki C kodumuzu(merhaba.c dosyası) make ile nasıl derleyeceğimizi anlatalım;

```
#include <stdio.h>
int main(){
    puts("Merhaba");
    return 0;
}
```

Makefile Nedir?

Makefile, make aracının nasıl çalışacağını belirten bir dosyadır. İçinde hedefler, bağımlılıklar ve komutlar bulunur. Örneğin, bir C programını derlemek için aşağıdaki gibi basit bir Makefile oluşturabilirsiniz. Makefile ve merhaba.c dosyası aynı konumda olmalıdır.

```
CC=gcc
CFLAGS=-I.
all: program
program: merhaba.o
        $(CC) -o program merhaba.o
merhaba.o: merhaba.c
        $(CC) -c merhaba.c $(CFLAGS)
clean:
        rm -f *.o program
```

merhaba dosyasından **program** adında bir ikili dosya oluşturmak için aşağıdaki komutlar Makefile ve merhaba.c dosyasının olduğu konumda çalıştırılır.

```
make
make install
```

Genellikle Makefile dosyası olmaz. Onun yerine **configure.ac** dosyası olur. **configure.ac** dosyasından **Makefile** dosyası elde etmek için, öncelikle **autoconf** ve **automake** araçlarının sisteminizde kurulu olması gerekmektedir.

autoreconf -fvi komutu çalıştırılarak configure dosyamızı üretir.Bu araçlar, yapılandırma dosyalarınızı işleyerek gerekli **Makefile** dosyalarını oluşturmanıza yardımcı olur.

configure komutunun devamında **--prefix** dışında başka parametreleride olabilir. Bu paramatreleri "Read.me" dosyası içerisinde bulunabilir veya **configure --help** komutu kaynak kodların olduğu konumda kullanılarak görülebilir. Bu kaynak kod aşağıdaki gibi derlenir.

```
$ autoreconf -fvi
$ ./configure --prefix=/usr
$ make
$ make install
```

2-cmake

CMake, projelerin derlenmesi ve yapılandırılması için kullanılan bir araçtır. Bir paketi CMake ile derlemek için genellikle aşağıdaki adımları izleriz:

İlk olarak, projenin kök dizininde bir CMakeLists.txt dosyası oluşturun. Ardından, CMake komutunu kullanarak projeyi yapılandırın ve derleyin. Örneğin:

```
    mkdir compile_packages
    cd compile_packages
    cmake ..
    make

Bu adımları takip ederek, CMake ile paketinizi başarıyla derleyebilirsiniz.
```

CMake'in esnekliği ve kullanım kolaylığı sayesinde paketlerin derlenmesi ve yapılandırılması oldukça kolay hale gelir.

Bu kaynak kod aşağıdaki gibi derlenir:

```
mkdir compile_packages
cd compile_packages
cmake ..
make
make install
```

3-meson

Meson, modern bir yapılandırma betik dili ve Ninja ise hızlı bir derleyici aracıdır. Bir paketi derlemek için öncelikle Meson ile yapılandırma dosyalarını oluşturmanız gerekir. Ardından, Ninja derleyici aracını kullanarak bu yapılandırmayı derleyebilirsiniz.

İlk olarak, projenizin dizinine gidin ve Meson ile yapılandırma dosyalarını oluşturun:

```
meson setup builddir --prefix=/usr
```

Sonra, oluşturulan builddir dizinine geçin ve Ninja ile derlemeyi başlatın:

```
ninja -C builddir
```

Bu adımları takip ederek Meson ve Ninja kullanarak paketinizi başarılı bir şekilde derleyebilirsiniz.

Bu kaynak kod aşağıdaki gibi derlenir:

```
# paketin yükleneceği konum
INSTALL_ROOT=/
meson setup builddir --prefix=/usr
ninja -C builddir
INSTALL_ROOT=$INSTALL_ROOT ninja -C builddir install
```

4-python

Python ile paket derlemek için genellikle **setuptools** ve **distutils** gibi kütüphaneler kullanılır. Öncelikle, projenizin kök dizininde bir **setup.py** dosyası oluşturmanız gerekir. Bu dosya, paketin yapılandırmasını ve gereksinimlerini tanımlar.

Örnek bir setup.py dosyası aşağıdaki gibi olabilir:

```
from setuptools import setup

setup(
    name='paket_adi',
    version='1.0',
    packages=['paket'],
    install_requires=[
        'bağımlılık_paketi',
    ],
)
```

Ardından, terminalde projenizin bulunduğu dizine giderek aşağıdaki komutu çalıştırarak paketi derleyebilirsiniz:

```
# yükleme konumu
INSTALL_ROOT=/
python3 -m compile_packages
python3 -m installer --destdir="$INSTALL_ROOT" dist/*.whl
```

Bu komut, paketi dist klasörüne derleyecektir. Artık paketiniz hazır ve dağıtıma uygun hale gelmiştir.

OpenRC

Openrc sistem açılışında çalışacak uygulamaları çalıştıran servis yöneticisidir.

Çalıştırılması

Openrc servis yönetiminin çalışması için boot parametrelerine yazılması gerekmektedir. **/boot/grub.cfg** içindeki **linux /vmlinuz init=/usr/sbin/openrc-init root=/dev/sdax** olan satırda **init=/usr/sbin/openrc-init** yazılması gerekmektedir. Artık sistem openrc servis yöneticisi tarafından uygulamalar çalıştırılacak ve sistem hazır hale getirilecek.

openrc Kullanımı

Servisleri etkinleştirip devre dışı hale getirmek için **rc-update** komutu kullanılır. Aşağıda **udhcpc** internet servisi örnek olarak gösterilmiştir. **/etc/init.d/** konumunda **udhcpc** dosyamızın olması gerekmektedir.

```
# servis etkinleştirmek için
rc-update add udhcpc boot
# servisi devre dışı yapmak için
rc-update del udhcpc boot
# Burada udhcpc servis adı, boot ise runlevel adıdır.
```

Elle **/etc/runlevels/** altında bulunan **boot**, **default**, **nonetwork**, **shutdown**, **sysinit** dizinlerine **/etc/init.d/** dizininin altındaki dosyaları **ln** komutuyla kısayol yaparsak **rc-update add** komutunun yaptığı görevi yapmış oluruz. Kısayolu silersek **rc-update del** komutunun görevini yapmış oluruz.

/etc/runlevels/ altında bulunan **boot**, **default**, **nonetwork**, **shutdown**, **sysinit** dizinler servis dosyalarımızın hangi sırayla çalışacağını belirleyen dizinlerdir. Mesela **boot** dizini ilk açılış sırasında çalışacak olan servis dosyalarının konulacağı dizindir.

Servisleri başlatıp durdurmak için ise **rc-service** komutu kullanılır.

```
rc-service udhcpc start
# veya şu şekilde de çalıştırılabilir.
/etc/init.d/udhcpc start
```

Servislerin durumunu öğrenmek için **rc-status** komutu kullanılır. Ayrıca sistemdeki servislerin sonraki açılışta hangisinin başlatılacağını öğrenmek için parametresiz olarak **rc-update** kullanabilirsiniz.

```
# şu an hangi servislerin çalıştığını gösterir
rc-status
# sonraki açılışta hangi servislerin çalışacağını gösterir
rc-update
```

Sistemi kapatmak veya yeniden başlatmak için **openrc-shutdown** komutunu kullanabilirsiniz.

```
openrc-shutdown -p 0     # kapatmak için
openrc-shutdown -r 0     # yeniden başlatmak için
```

Servis Dosyası

Openrc servis dosyaları basit birer **bash** betiğidir. Bu betikler **openrc-run** komutu ile çalıştırılır ve çeşitli fonksiyonlardan oluşabilir. Servis dosyaları **/etc/init.d** içerisinde bulunur. Servisleri ayarlamak için ise **/etc/conf.d** içerisine aynı isimle ayar dosyası oluşturabiliriz.

Çalıştırılacak komut, komut parametreleri ve **pidfile** dosyamızı aşağıdaki gibi belirtebiliriz.

```
description="Ornek servis"
command=/usr/bin/ornek-servis
command_args=--parametre
pidfile=/run/ornek-servis.pid
```

Bununla birlikte **start**, **stop**, **status**, **reload**, **start_pre**, **stop_pre** gibi fonksiyonlar da yazabiliriz.

```
start(){
    ebegin "Starting ${RC_SVCNAME}"
    start-stop-daemon --start --pidfile "/run/servis.pid" --exec /usr/bin/ornek-servis --parametre
}
```

Servis bağımlılıklarını belirtmek için ise **depend** fonksiyonu kullanılır.

```
depend() {
  need localmount
  after dbus
}
```

Qemu Kullanımı

qemu açık kaynaklı Virtual Box, Vmware benzeri sanallaştırma aracıdır.

Sisteme Kurulum

```
sudo apt update
sudo apt install qemu-system-x86  qemu-utils
```

qemu Kullanımı

```
# 30GB disk oluşturuldu.
qemu-img create disk.img 30G
qemu-system-x86_64 --enable-kvm -hda disk.img -m 2G -cdrom etahta.iso
# qemu-system-x86_64 --enable-kvm -hda disk.img -m 2G #sadece disk ile çalıştırılıyor
# qemu-system-x86_64 -m 2G -cdrom etahta.iso          #sadece iso dosyası ile çalıştırma
```

Sistem Hızlandırılması

--enable-kvm eğer sistem disk ile çalıştırıldığında bu parametre eklenmezse yavaş çalışacaktır.

Boot Menu Açma

Sistemin diskten mi imajdan mı başlayacağını başlangıçta belirlemek için boot menu gelmesini istersek aşağıdaki gibi komut satırına seçenek eklemeliyiz.

```
qemu-system-x86_64 --enable-kvm -cdrom distro.iso -hda disk.img -m 4G -boot menu=on
```

Uefi kurulum için:

sudo apt-get install ovmf

qemu-system-x86_64 --enable-kvm -bios /usr/share/ovmf/OVMF.fd -cdrom distro.iso -hda disk.img -m 4G -boot menu=on

qemu Host Erişimi:

İstemci bilgisayar ip'si:10.0.2.15 ve ana bilgisayar 10.0.0.2 olarak ayarlıyor.

vmlinuz ve initrd

qemu ile vmlinuz ve initrd.img dosyaları iso olmadan test edilebilir.

```
qemu-system-x86_64 --enable-kvm -kernel /boot/vmlinuz-5.17 -initrd /home/deneme/initrd.img -append "quiet" -m 512m
```

qemu Terminal Yönlendirmesi

```
qemu-system-x86_64 --enable-kvm -kernel vmlinuz -initrd initrd.img -m 3G -serial stdio -append "console=ttyS0"
```

Diskteki Sistemin Açılışını Terminale Yönlendirme

```
qemu-system-x86_64 -nographic -kernel boot/vmlinuz -hda disk.img -append console=ttyS0
```

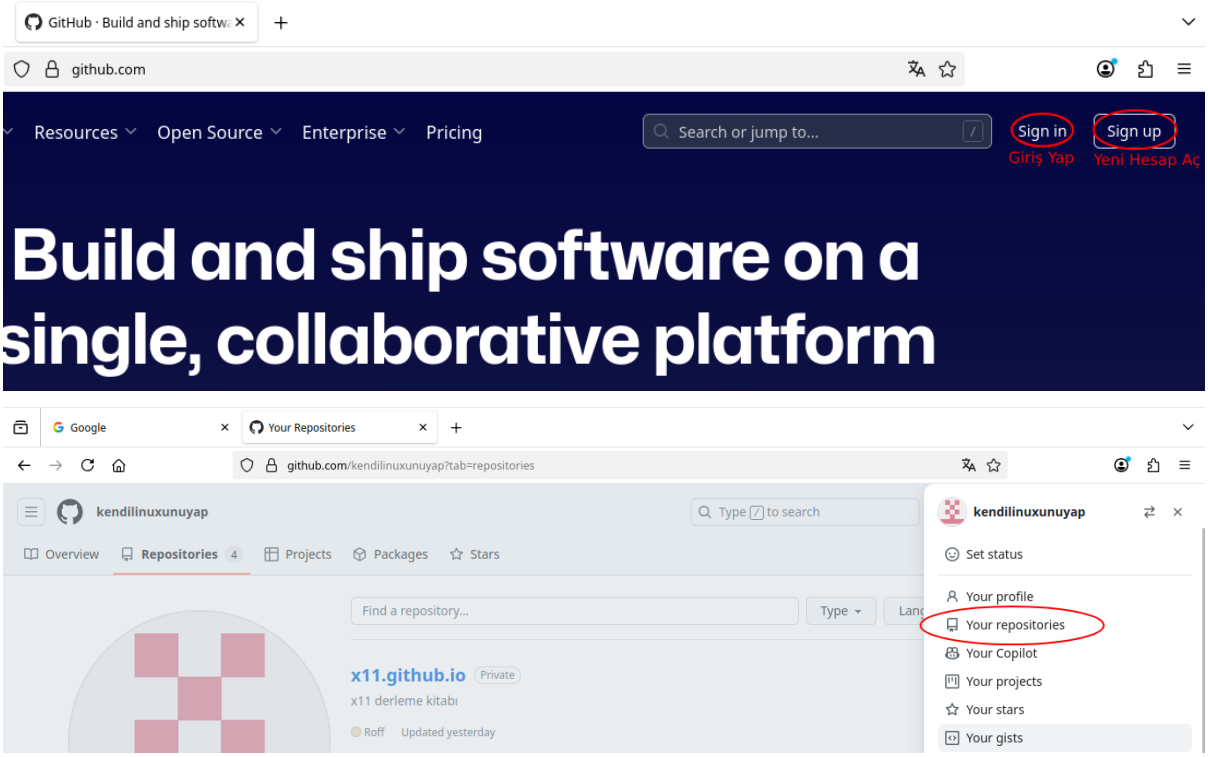
Kaynak: | https://www.ubuntubuzz.com/2021/04/how-to-boot-uefi-on-qemu.html

github

GitHub üzerinde yeni bir depo açmak oldukça basit bir işlemdir. Aşağıdaki adımları takip ederek hızlıca kendi deponuzu oluşturabilirsiniz:

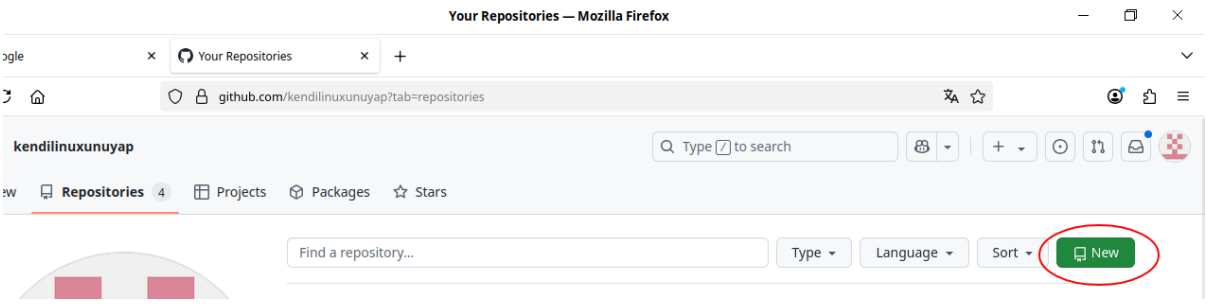
GitHub Hesabınıza Giriş Yapın

GitHub ana sayfasına gidin ve hesabınıza giriş yapın. Eğer bir hesabınız yoksa, öncelikle bir hesap oluşturmalısınız.



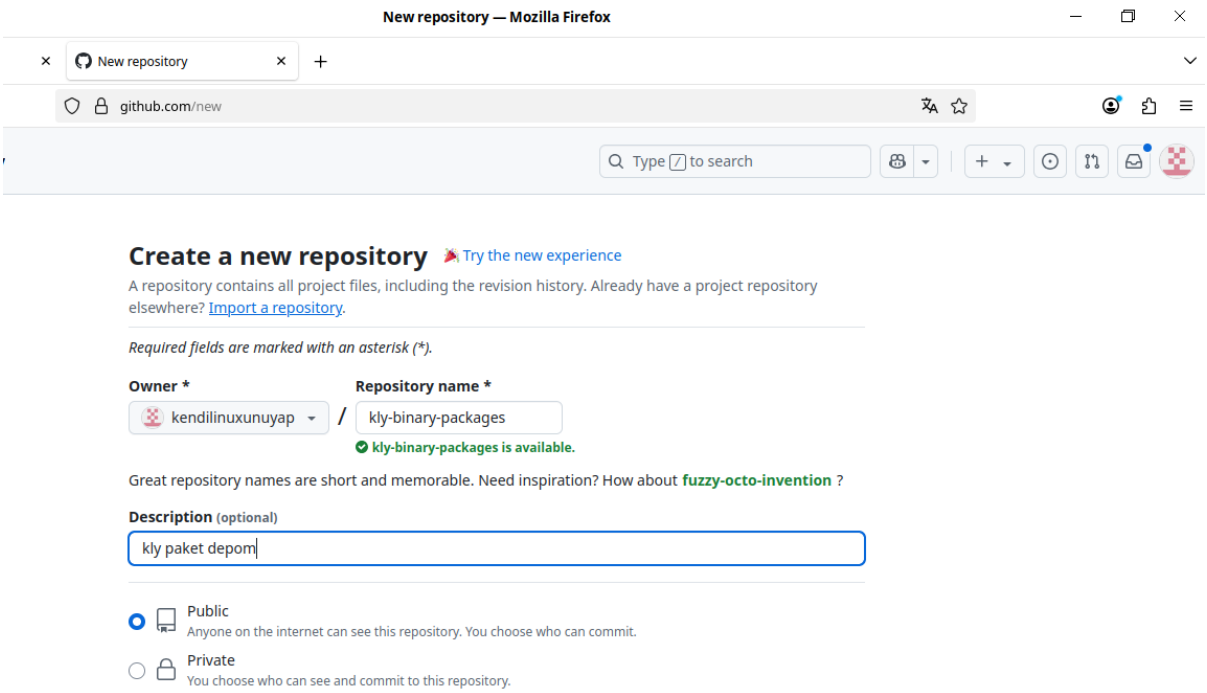
Yeni Depo Oluşturma

Sağ üst köşede bulunan "+" simgesine tıklayın ve "New repository" seçeneğini seçin.



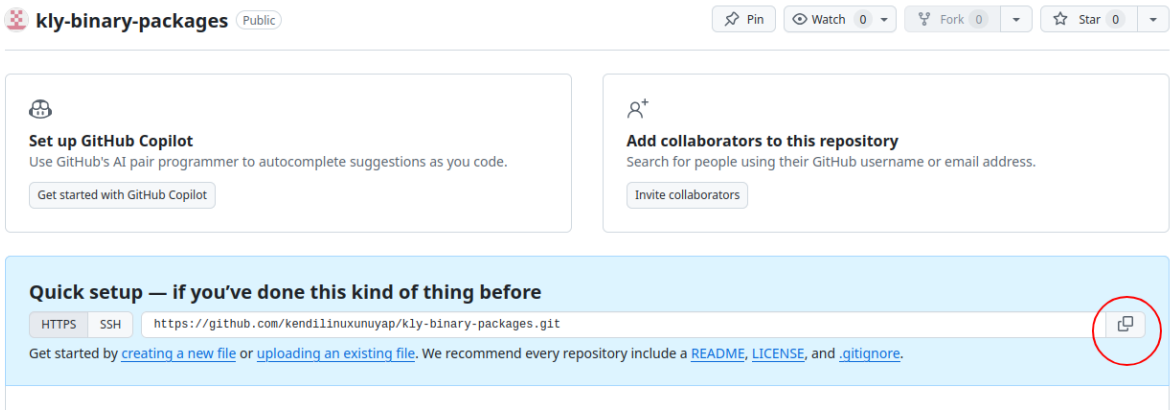
Depo Bilgilerini Girin

Açılan sayfada, depo adını (repository name) ve isteğe bağlı olarak bir açıklama (description) girin. Depo özel (private) veya herkese açık (public) olarak ayarlanabilir.



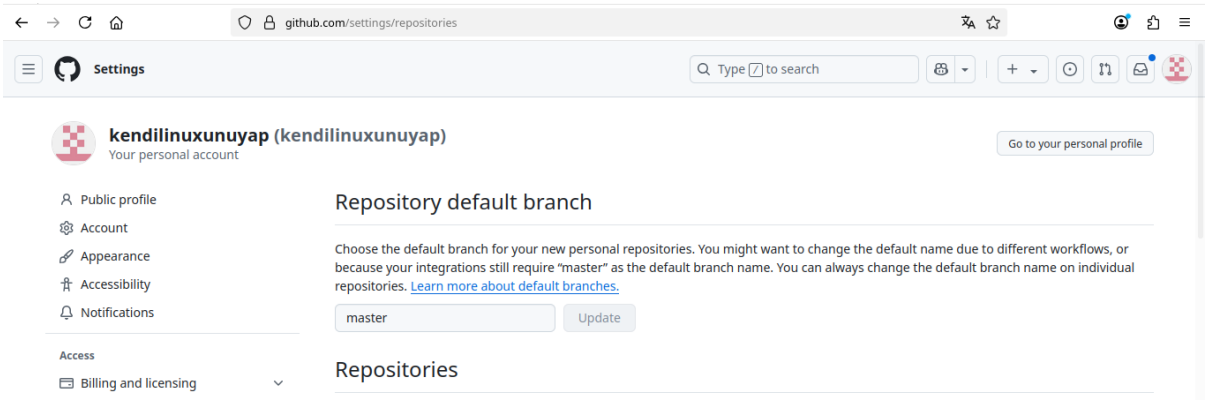
İlk Dosyayı Oluşturma

"Initialize this repository with a README" seçeneğini işaretleyerek, depo oluşturulduğunda otomatik olarak bir README dosyası oluşturabilirsiniz.

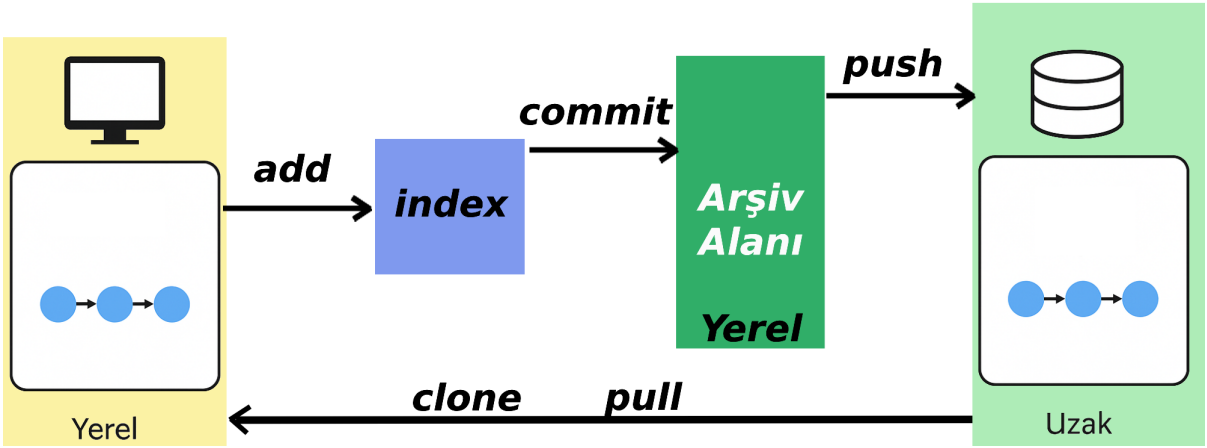


github Varsayılan Dal Ayarı:

githubda varsayılan olarak eski sürümlerde master, yeni sürümlerde main kullanılmaktadır. Projelerimizde ve burada kullanılan yapılarda **master** kullanıldığı için aşağıda görülduğü gibi varsalılan dalı **master** yapıyoruz.



github komut Kullanımı



github'ın çalışma mantığı yukarıda verilen resimde görüldüğü gibidir. Komutları kullanırken resimdeki gibi işlemleri yapmalıyız.

github'a göndermek için; **add --> commit --> push** kullanmalıyız.

github'dan indirmek için; **clone veya pull** kullanmalıyız.

Sık Kullanılan github Komutları

Depoyu Yerele indirme(Clone)

```
git clone https://github.com/kullaniciadi/depoadi.git
```

- **Yerel Depoda Dosya Ekleme:**

```
git add .
```

- **Yerel Depoda Değişiklik Etiketi Yapma:**

```
git commit -m "ilk adım"
```

- **Yerel Depodaki Bilgileri Guthuba Gönderme:**

```
git push origin master
```

- **Yerel Depodaki Bilgileri Guthuba Gönderme Reddedilirse:**

```
git push origin master --force
```

- **Yerelde github'daki Depoyu clone Yapmadan Oluşturma:**

```
cd proje
git init
git config --global user.name "name"
git config --global user.email "name@gmail.com"
git add README.md
git add .
git commit -m "first commit"
git remote -v      # push ve pull yapılacak adresleri görmek için kullanılır
git remote add origin https://github.com/userName/repoName.git
git push -u origin master
```

Dal(Branch):

Dal projenin birden fazla kişi ile yapılmasında veya yeni özellikler eklenmek istediğinde projenin bir kopyası ile çalışma gerektirir. Aşağıda dal işlemleri için komutlar verilmiştir.

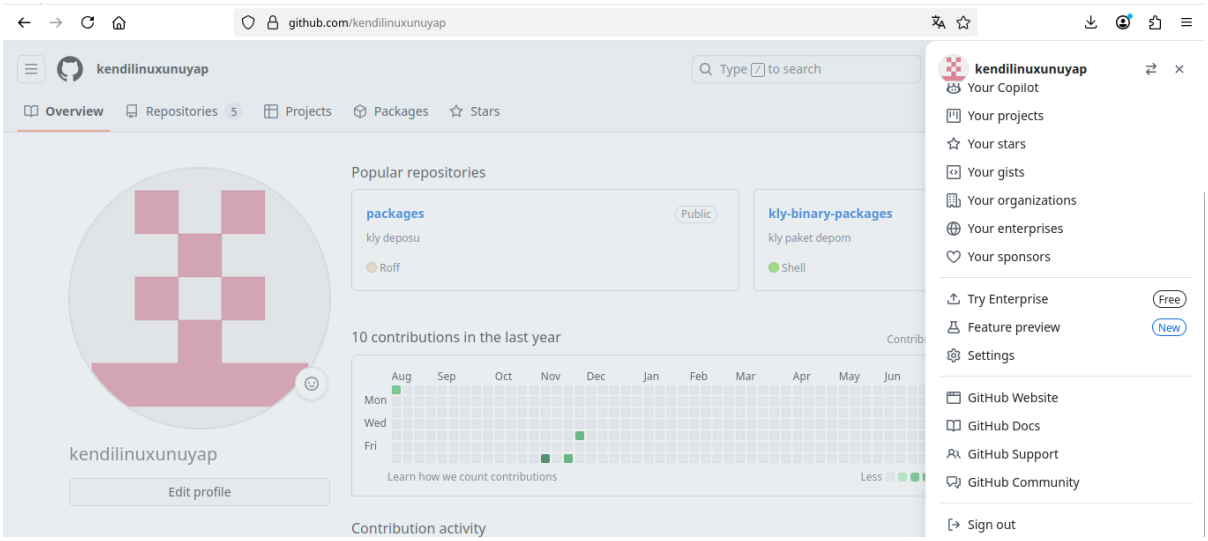
Yeni Dal Oluşturmak, Seçmek ve Yeni Dalı github'a Göndermek:

```
# Dalları Görmek kullanılır Seçili olan dalın rengi farklı olur ve önünde * olur
git branch
# Dal oluşturmak
git checkout yeni
# Yeni dalı uzak adrese göndermek
git push --force origin master komutu  yerine
# Uzak adresimizde master dalı dışında yeni dalımızda oluşacaktır...
git push --force origin yeni komutunu veriyoruz.
```

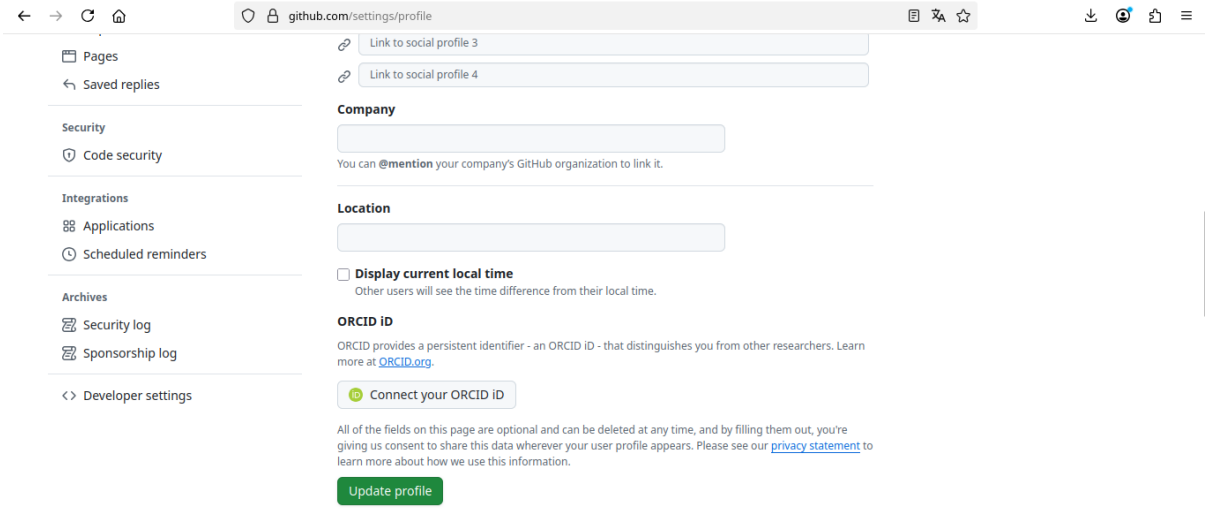
github token Oluşturma Kullanma

github kullanırken dosya gondermek(**commit**) için kullanıcı adı ve parola ister. Burada hesap parolası yerine **token** kullanılır. Yeni bir **token** için aşağıdaki işlem adımlarını yapmalıyız.

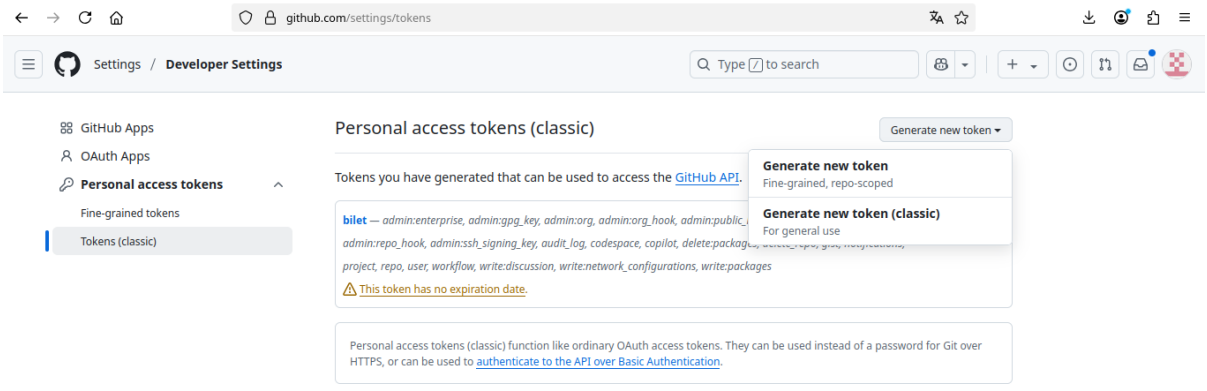
Settings Seçilir;



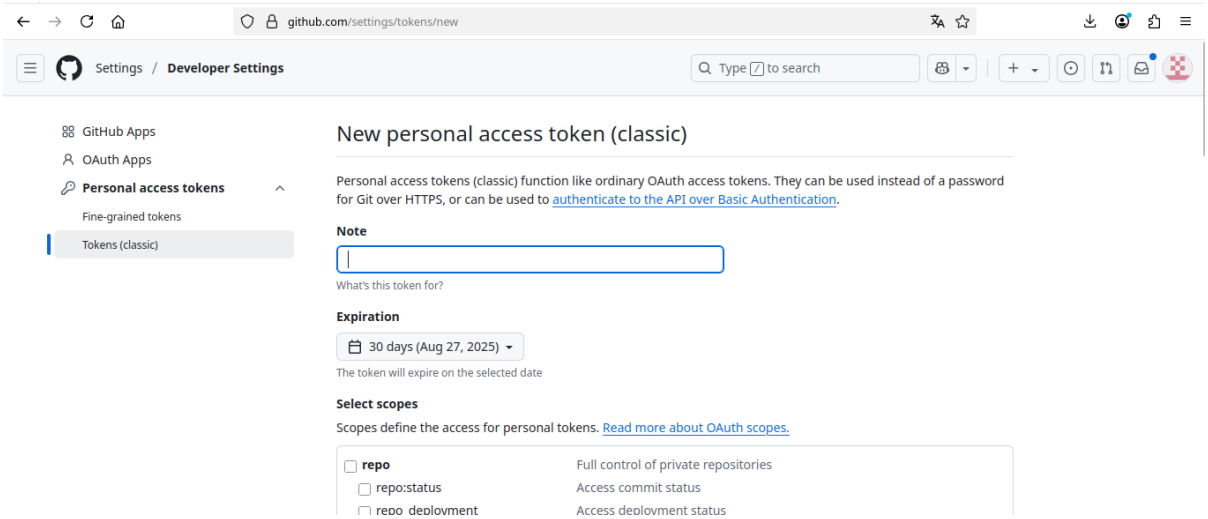
Developper Settings Seçilir;



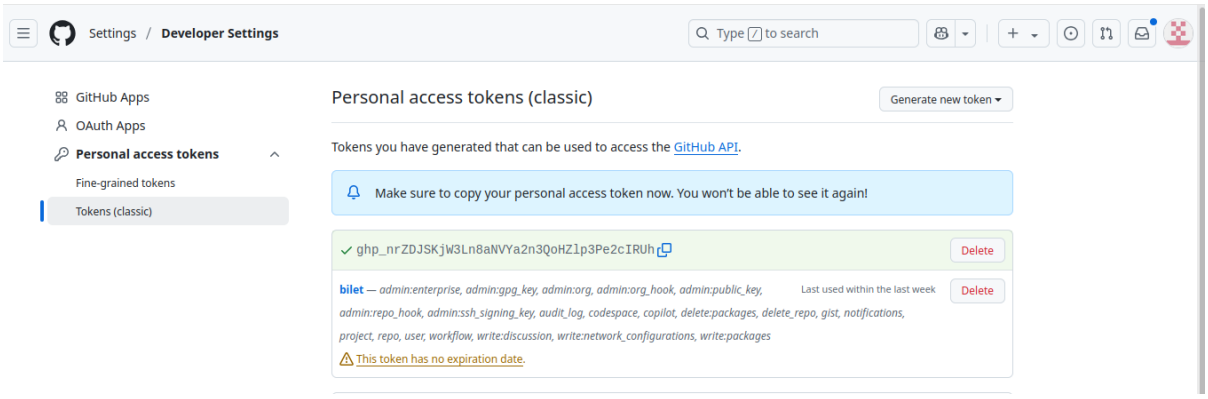
Generate New Token Seçilir;



Erişim yapabileceği alanlar Seçilir;



Token kullanmak üzere kullanmak üzere saklanır. Bu ekrandan sonra sadece silebiliriz. Göremeyiz kopyalayamayız.



elogind Yapılandırması

Bu belge, **systemd kullanmayan** sistemlerde (OpenRC tabanlı) elogind login manager'ın kurulumu, yapılandırılması ve gerekli kullanıcı/PAM/agetty ayarlarını anlatır.

1. Gerekli Kullanıcı ve Grup Tanımları

elogind servisi için sistemde gerekli kullanıcı ve gruplar oluşturulur:

```
groupadd -g 190 systemd-journal
groupadd -g 191 elogind
useradd -r -s /sbin/nologin -g elogind -u 191 elogind
```

Açıklama:

- *systemd-journal* grubu, journal log erişimi için kullanılır.
- *elogind* kullanıcı ve grubu, servis için ayrılmıştır.
- */sbin/nologin* ile bu kullanıcıyla doğrudan login engellenir.
- *-r* parametresi sistemi kullanıcı oluşturur, yani normal login için kullanılmaz.

2. OpenRC Hizmet Dosyası (/etc/init.d/elogind)

OpenRC ile *elogind* servisini yönetmek için basit bir init script örneği:

```
#!/sbin/openrc-run
description="elogind login manager"
command=/usr/lib/elogind/elogind
pidfile=/run/elogind.pid

depend() {
    need dbus
    after dbus
}
```

Açıklama:

- *command*: elogind daemonunu başlatır.
- *pidfile*: servis PID bilgisini saklar.
- *depend()*: servis başlatılmadan önce D-Bus'ın aktif olmasını garanti eder.

3. Başlatma ve OpenRC'ye Ekleme

```
rc-update add elogind default
rc-service elogind start
```

Açıklama:

- *rc-update add*: Servisi boot sırasında otomatik başlatmaya ekler.
- *rc-service start*: Servisi hemen başlatır.
- **Önemli Kontrol:** elogind'in çalışıp çalışmadığını kontrol etmek için:

```
loginctl list-sessions
```

- Eğer aktif oturumlar listeleniyorsa, elogind başarıyla çalışıyor demektir.

4. PAM, Passwd ve Group Ayarları

/etc/nsswitch.conf Örneği:

```
passwd:      files
group:       files
shadow:      files
```

Açıklama:

- Kullanıcı ve grup bilgileri *files* (yerel /etc/passwd ve /etc/group) üzerinden çözülür.

/etc/pam.d/system-auth Örneği:

```
auth        required      pam_unix.so
account     required      pam_unix.so
password    required      pam_unix.so
session     required      pam_unix.so
session     include       elogind-user
```

> Note
>
> **`session include elogind-user` satırı mutlaka ekli olmalıdır.** Bu satır, kullanıcı oturumlarının elogind ile doğru şekilde yönetilmesini sağlar. Eğer yoksa manuel olarak eklenmelidir:
>
> ```
> sed -i "/elogind-user/d" /etc/pam.d/system-auth
> echo -e "\nsession    include    elogind-user" >> /etc/pam.d/system-auth
> ```

5. OpenRC Agetty Ayarı

Bazı durumlarda *agetty*, varsayılan olarak */bin/login* kullanmaz ve bu nedenle elogind kullanıcı oturumunu başlatamaz.

Ayar:

```
sed -i "/agetty_options/d" /etc/conf.d/agetty
echo -e "\nagetty_options=\"-l /usr/bin/login\"" >> /etc/conf.d/agetty
```

Açıklama:

- *-l /usr/bin/login* ile agetty doğru login programını çağırır.
- Bu ayar, sanal konsollardan kullanıcı girişlerinin doğru şekilde yapılmasını garanti eder.

6. Diğer Öneriler ve Notlar

- *elogind*, sistemde */run/systemd* gibi dizinler oluşturur; böylece bazı uygulamalar systemd'ye ihtiyaç duymadan çalışabilir.
- Polkit yetkilendirme sorunları genellikle kullanıcıların *wheel* grubuna dahil olmamasıyla ilişkilidir.
- OpenRC + elogind ile systemd'ye gerek kalmadan oturum yönetimi sağlanabilir.

Kaynaklar:

```
- elogind resmi dokümanı: https://www.freedesktop.org/wiki/Software/systemd/elogind/
- OpenRC belgeleri: https://wiki.gentoo.org/wiki/OpenRC
- PAM belgeleri: https://www.linux-pam.org/
- Polkit belgeleri: https://www.freedesktop.org/wiki/Software/polkit/
```

D-Bus Yapılandırması

Bu belge, **systemd kullanmayan** sistemlerde (OpenRC tabanlı) D-Bus servisinin kurulumu, yapılandırılması ve başlatılmasını anlatır. D-Bus, Linux ortamında uygulamalar arasında mesajlaşma sağlayan bir IPC (Inter-Process Communication) sistemidir.

Adım 1: Kullanıcı ve Grup Oluşturma

```
echo "messagebus:x:109:" >> /etc/group
echo "messagebus:x:103:109::/nonexistent:/usr/sbin/nologin" >> /etc/passwd
```

Adım 2: Sistem UUID ve machine-id Ayarı

```
if [ ! -f /etc/machine-id ] ; then
    dbus-uuidgen --ensure=/etc/machine-id
fi
```

Açıklama:

```
- `/etc/machine-id` dosyası sistemin benzersiz kimliğini tutar.
- `dbus-uuidgen --ensure` komutu, dosya yoksa oluşturur, varsa korur.
- Bu kimlik, D-Bus servislerinin ve bazı uygulamaların doğru çalışması için zorunludur.
```

Adım 3: dbus-daemon-launch-helper İzin Kontrolü

```
if busybox ls /bin/su -la | grep "^...s" >/dev/null ; then
    chmod 4755 /usr/libexec/dbus-daemon-launch-helper
fi
```

Açıklama:

```
- `dbus-daemon-launch-helper`, kullanıcı bazlı D-Bus daemonlarını başlatır.
- SUID bitinin (4) set edilmesi, root yetkisi ile çalışmasını sağlar.
```

Adım 4: D-Bus Yapılandırma Yenileme

```
dbus-send --system --type=method_call --dest=org.freedesktop.DBus / \
    org.freedesktop.DBus.ReloadConfig >/dev/null 2>&1 || :
```

Açıklama:

```
- Bu komut, D-Bus sistem servisine yeni yapılandırmaları yüklemesini söyler.
- `--system` parametresi sistem bus'ını hedef alır.
- Hata oluşursa komut sessizce geçer (`|| :`).
```

Adım 5: D-Bus Servisini Başlatma (OpenRC)

D-Bus için OpenRC init script'i */etc/init.d/dbus* şu mantıkla çalışır:

```
#!/sbin/openrc-run
#-----------------------------------------------------------------------------------------------
name="System Message Bus"
description="An IPC message bus daemon"

extra_started_commands="reload"

supervisor=supervise-daemon
command="/usr/bin/dbus-daemon"
command_args="--system --nofork --nopidfile --syslog-only ${command_args:-}"
command_background="yes"
pidfile="/run/$RC_SVCNAME.pid"

depend() {
    need localmount
    after bootmisc
}

start_pre() {
    mkdir -p /run/dbus
    /usr/bin/dbus-uuidgen --ensure=/etc/machine-id
}

stop_post() {
    [ ! -S /run/dbus/system_bus_socket ] || rm -f /run/dbus/system_bus_socket
}

reload() {
    ebegin "Reloading $name configuration"
    /usr/bin/dbus-send --print-reply --system --type=method_call \
            --dest=org.freedesktop.DBus \
            / org.freedesktop.DBus.ReloadConfig > /dev/null
    eend $?
}
```

Açıklama:

```
- `supervise-daemon`: OpenRC'nin daemon'u denetlemesini sağlar.
- `command` ve `command_args`: D-Bus daemonunu sistem bus modunda başlatır.
- `start_pre()`: Servis başlamadan önce gerekli dizini oluşturur ve machine-id'yi doğrular.
- `stop_post()`: Servis durduğunda socket dosyasını temizler.
- `reload()`: Yapılandırma değişikliklerini uygulamak için D-Bus'a mesaj yollar.
- `depend()`: Servisin başlatılması için gerekli bağımlılıklar (localmount, bootmisc) belirtilir.
- `pidfile`: Servisin PID bilgisini tutar.
```

Servis Dosyasının Yaptıklar:

```
- D-Bus için sistem kullanıcı ve grup oluşturulur.
- machine-id ve gerekli izinler ayarlanır.
- OpenRC init script'i `/etc/init.d/dbus`, servisin başlatılmasını, durdurulmasını ve yeniden yüklenmesini yönetir.
- `rc-service dbus start` ile sistemde mesajlaşma servisi aktif olur.
```

Kaynaklar:

```
- D-Bus resmi sitesi:https://www.freedesktop.org/wiki/Software/dbus/
- D-Bus belgeleri:https://dbus.freedesktop.org/doc/dbus/
- OpenRC belgeleri:https://wiki.gentoo.org/wiki/OpenRC
- dbus-daemon ve dbus-send man sayfaları:https://manpages.debian.org/dbus-user-session
```

polkit Yetkilendirme

Bu kılavuz, lxsession-logout penceresinde "Kapat" ve "Yeniden Başlat" işlemlerinin parola doğrulaması açılmasına rağmen başarısız olması sorununu çözer. Amaç: *wheel* grubundaki kullanıcıların GUI'den **poweroff/reboot** yapabilmesini sağlamak.

Temel Ayar Kontrolü

1. Elogind çalışıyor mu?

```
rc-status | grep -E 'elogind|dbus'
rc-service elogind status
```

2. Polkit aracı (agent) çalışıyor mu?

```
pgrep -fa lxpolkit || pgrep -fa polkit-gnome-authentication-agent-1
```

3. Kullanıcı gerçekten *wheel* grubunda mı?

```
id $USER
```

Çözüm 1 — *wheel* Grubunu “Yönetici” Olarak Tanımlama

Polkit’e *wheel* grubunun “yönetici kimliği (AdminIdentity)” olduğunu söyleyin. Bu, yönetici doğrulaması gerektiren işlemlerde *wheel* üyesinin **kendi parolasıyla** yetki almasını sağlar.

1. Dosyayı oluşturun: `/etc/polkit-1/rules.d/40-wheel-admin.rules`

```
/* Mark wheel group as administrators (use own password) */
polkit.addAdminRule(function(action, subject) {
    if (subject.isInGroup("wheel")) {
        return ["unix-group:wheel"];
    }
});
```

2. Dosya izinleri:

```
chown root:root /etc/polkit-1/rules.d/40-wheel-admin.rules
chmod 0644 /etc/polkit-1/rules.d/40-wheel-admin.rules
```

3. Tekrar oturum açın ve `lxsession-logout` üzerinden deneyin.

Çözüm 2 — Sadece Güç Eylemlerine *wheel* için “YES” Ver

Sistemde “admin” kavramını genişletmek istemiyorsanız, yalnızca poweroff/reboot eylemlerine *wheel* için açıkça izin verin.

1. Dosyayı oluşturun: `/etc/polkit-1/rules.d/49-login1-power-wheel.rules`

```
/* Allow wheel to poweroff/reboot (elogind login1 actions) */
polkit.addRule(function(action, subject) {
    if (!subject.isInGroup("wheel"))
        return;

    var a = action.id;
    if (a == "org.freedesktop.login1.power-off" ||
        a == "org.freedesktop.login1.power-off-multiple-sessions" ||
        a == "org.freedesktop.login1.reboot" ||
        a == "org.freedesktop.login1.reboot-multiple-sessions") {
        return polkit.Result.YES;
    }
});
```

2. İzinler:

```
chown root:root /etc/polkit-1/rules.d/49-login1-power-wheel.rules
chmod 0644 /etc/polkit-1/rules.d/49-login1-power-wheel.rules
```

3. Tekrar oturum açın ve `lxsession-logout` üzerinden deneyin.

Önemli Not:

Sisteminizde polkit kullanıyorsanız ve masaüstü ortamı **LXDE** ise **lxpolkit** devre dışı bırakılmalıdır. Kaldırma işlemini **lxsession** otomatik başlatma seçeneklerinden **lxpolkit** kaldırılarak yapılır. Kaldırılmaması durumunda birden fazla yetkilendirme çalışıyor uyarısı alırız.

Kaynaklar:

```
- OpenRC Resmi Belgeler:        https://wiki.gentoo.org/wiki/OpenRC
- Polkit Resmi Dokümantasyon:   https://www.freedesktop.org/software/polkit/docs/latest/
- Arch Wiki — Polkit:   https://wiki.archlinux.org/title/Polkit
- Arch Wiki — Elogind:  https://wiki.archlinux.org/title/Elogind
- LXDE Resmi Wiki:      https://wiki.lxde.org/en/Main_Page
- Debian Wiki — Polkit: https://wiki.debian.org/PolicyKit
- Polkit kural örnekleri (wheel group): https://wiki.archlinux.org/title/Polkit#Bypass_password_prompt
```

pkexec, /etc/shells ve Polkit İlişkisi

Bu belge, pkexec komutunun çalışması için gerekli olan `/etc/shells` bağlantısını, Polkit ile olan ilişkisini ve doğru yapılandırma adımlarını açıklar.

1 — Temel İlişki

- **pkexec**, bir komutu yönetici yetkileri ile çalıştırmak için kullanılır.
- Arka planda **Polkit** (PolicyKit) kullanarak yetkilendirme yapar.
- Yetkilendirme sırasında **PAM (Pluggable Authentication Modules)** kullanılır.
- Birçok PAM yapılandırması, kullanıcının giriş kabuğunun **/etc/shells** içinde listelenmiş olmasını zorunlu kılar.

> Note
>
> Eğer kullandığınız kabuk (örn. `/bin/bash`, `/bin/zsh`, `/usr/bin/fish`) `/etc/shells` içinde yoksa pkexec ile kimlik doğrulama başarısız olabilir.

2 — /etc/shells Dosyasını Kontrol Etme

Geçerli kabuğunuz:

```
echo $SHELL
```

Kabuğun `/etc/shells` içinde olup olmadığını kontrol edin:

```
grep "$(basename $SHELL)" /etc/shells
```

Eksikse ekleyin:

```
sudo sh -c 'echo /bin/bash >> /etc/shells'
```

> Warning
>
> Güvenlik sebebiyle yalnızca gerçek kabuk programlarının yolunu ekleyin. Rastgele betikler veya çalıştırılabilir dosyalar eklenmemelidir.

3 — Polkit ile Bağlantı

- pkexec çalışırken **Polkit** üzerinden yetki ister.
- Polkit kuralları ile hangi kullanıcı veya grubun yetki alabileceğini belirleyebilirsiniz.
- Eğer kullanıcı grubunuz Polkit tarafından **admin** olarak tanınmıyorsa, pkexec ile çalıştırma yetkiniz olmayabilir.

4 — Wheel Grubunu Yönetici Olarak Tanımlama

1) Dosya oluşturun:

```
/etc/polkit-1/rules.d/40-wheel-admin.rules
```

İçeriği:

```
/* Mark wheel group as administrators (use own password) */
polkit.addAdminRule(function(action, subject) {
    if (subject.isInGroup("wheel")) {
        return ["unix-group:wheel"];
    }
});
```

2) Dosya izinlerini ayarlayın:

```
sudo chown root:root /etc/polkit-1/rules.d/40-wheel-admin.rules
sudo chmod 0644 /etc/polkit-1/rules.d/40-wheel-admin.rules
```

3) Kullanıcınızı wheel grubuna ekleyin:

```
sudo usermod -aG wheel $USER
```

4. Oturumu kapatıp tekrar açın.

5 — Test

1) pkexec komutunu çalıştırın:

```
pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY xterm
```

2. Parola soruluyorsa ve yetkilendirme başarılı oluyorsa ayarlarınız tamamdır.

6 — Özet

- pkexec → Polkit kullanarak yetkilendirme yapar.
- Polkit → PAM üzerinden kimlik doğrular.
- PAM → Kullanıcı kabuğunun `/etc/shells` içinde olmasını bekler.
- `/etc/shells` → Geçerli kabuğunuz burada yoksa pkexec başarısız olur.
- Polkit kuralları ile hangi kullanıcı/grupların yetki alacağı belirlenir.

Kaynaklar:

```
- Polkit Dokümantasyonu: https://www.freedesktop.org/software/polkit/docs/latest/
- Arch Wiki — Polkit: https://wiki.archlinux.org/title/Polkit
- Arch Wiki — Sudo: https://wiki.archlinux.org/title/Sudo
- PAM /etc/shells: https://linux.die.net/man/5/shells
```

glib-compile-schemas Kullanımı

`glib-compile-schemas` komutu, GLib kütüphanesinin GSettings yapılandırma sisteminde kullanılan `*.gschema.xml` dosyalarını ikili (binary) biçime derlemek için kullanılır.

Derleme işlemi sonunda `gschemas.compiled` adlı tek bir dosya oluşur. Bu dosya, uygulamaların GSettings üzerinden yapılandırma okuma/yazma işlemlerini hızlı ve verimli şekilde yapmasını sağlar.

Temel Görev

- `*.gschema.xml` → `gschemas.compiled` dönüşümünü yapmak.
- GSettings yapılandırma şemalarının sistem tarafından tanınmasını sağlamak.
- XML dosyalarının doğrudan okunması yerine, önceden derlenmiş ikili dosyadan hızlı erişim imkânı sunmak.

Ne Zaman Kullanılır?

1. **Yeni bir GSettings şeması eklendiğinde:** Sisteme yeni bir paket veya uygulama kurulduğunda ve bu uygulama kendi `*.gschema.xml` dosyalarıyla birlikte geliyorsa, bu dosyaların derlenmesi gerekir.
2. **Mevcut bir şema değiştirildiğinde**
3. **Sistem yükleme veya imaj oluşturma sürecinde:** Özel bir Linux imajı hazırlanırken (ör. distro yapımı) şemalar eklendikten sonra çalıştırılmalıdır.
4. **Manuel şema kurulumu yapıldığında:** Paket yöneticisi dışında, elle `/usr/share/glib-2.0/schemas/` dizinine XML dosyası kopyalandığında.

Kullanım Şekli

```
sudo glib-compile-schemas /usr/share/glib-2.0/schemas/
```

Açıklama:

- `/usr/share/glib-2.0/schemas/` GLib tarafından kullanılan ana şema dizinidir. Burada tüm paketlere ait `*.gschema.xml` dosyaları bulunur.
- Komut, bu dizindeki tüm XML şemalarını tarar ve tek bir `gschemas.compiled` dosyası üretir.

Dikkat Edilmesi Gerekenler

- Komutun, şemaların bulunduğu dizin üzerinde yazma iznine sahip bir kullanıcı (genellikle `root`) tarafından çalıştırılması gerekir.
- `*.gschema.xml` dosyalarında sözdizimi hatası varsa derleme başarısız olur.
- Derleme sonrası `gschemas.compiled` dosyasının mevcut ve güncel olduğundan emin olun.

Kaynaklar:

```
- GLib GSettings belgeleri:     https://developer.gnome.org/gio/stable/GSettings.html
- glib-compile-schemas kılavuz sayfası: https://manpages.debian.org/glib-compile-schemas
```

gdk-pixbuf-query-loaders Kullanımı

`gdk-pixbuf-query-loaders` komutu, `gdk-pixbuf` kütüphanesinin kullanabileceği tüm resim yükleyicilerini (image loaders) tarar ve bunların bilgilerini derleyerek bir yapılandırma dosyası (`loaders.cache`) oluşturur.

Bu komut, özellikle GNOME ve GTK tabanlı uygulamaların farklı resim formatlarını (PNG, JPEG, SVG, GIF vb.) tanıyabilmesi için gereklidir.

Temel Görev

- Sistem üzerinde kurulu **gdk-pixbuf loader modüllerini** bulur.
- Bu modüllerin yeteneklerini (desteklenen dosya türleri, MIME tipleri vb.) listeler.
- Bilgileri ikili önbellek dosyası olan `loaders.cache` içine yazar.
- Böylece uygulamalar, her çalıştırıldığında modül taraması yapmak yerine bu önbellekten hızlıca bilgi alır.

Ne Zaman Kullanılır?

1. **Yeni image loader eklendiğinde:** Örneğin, SVG desteği için `librsvg` kurulduğunda veya başka bir formatı destekleyen eklenti eklendiğinde.
2. **gdk-pixbuf güncellendiğinde:** Kütüphane sürümü değiştiğinde, modül yolları değişebileceği için `loaders.cache` dosyasının yeniden oluşturulması gerekir.
3. **Sistem imajı hazırlanırken:** Özel bir Linux dağıtımı yapılırken, kök dosya sisteminde tüm loader'ların kayıtlı olduğundan emin olmak için.
4. **Elle modül kurulumu yapıldığında:** Paket yöneticisi dışında manuel olarak bir loader modülü kopyalandığında veya derlendiğinde.

Kullanım Şekli

```
sudo gdk-pixbuf-query-loaders --update-cache
```

Açıklama:

```
- ``--update-cache`` Otomatik olarak tüm loader'ları tarar ve sonuçları ``loaders.cache`` dosyasına yazar.
- Varsayılan cache dosyası konumu: ``/usr/lib/x86_64-linux-gnu/gdk-pixbuf-2.0/2.10.0/loaders.cache`` (Konum, mimari ve dağıtıma göre değişebilir.)
- ``gdk-pixbuf-query-loaders`` komutu **stdout** çıktısı verir; eğer ``--update-cache`` kullanılmazsa bu çıktı elle bir dosyaya yönlendirilmelidir:
```

```
gdk-pixbuf-query-loaders > /usr/lib/.../loaders.cache
```

Dikkat Edilmesi Gerekenler

- Komutun, cache dosyasının bulunduğu dizine yazma izni olan kullanıcı (genellikle `root`) tarafından çalıştırılması gerekir.
- Modüller doğru yüklenmiyorsa veya bazı formatlar açılamıyorsa, cache dosyasının güncel olduğundan emin olun.
- Yanlış mimarideki (32-bit/64-bit) loader modülleri eklenirse hatalar oluşabilir.

Kaynaklar:

```
- GDK-Pixbuf Belgeleri: https://developer.gnome.org/gdk-pixbuf/stable/
- gdk-pixbuf-query-loaders kılavuz sayfası:     https://manpages.debian.org/gdk-pixbuf-query-loaders
```

user-dirs

user-dirs, Linux işletim sistemlerinde kullanıcıların özel klasörlerini içeren bir sistemdir. Bu klasörler genellikle "Masaüstü", "Belgeler", "İndirilenler" gibi isimlerle tanımlanır ve kullanıcıların dosyalarını düzenlemelerini kolaylaştırır.

user-dirs klasörünü kaldırmak için aşağıdaki adımları izleyebilirsiniz:

Terminali açın ve aşağıdaki komutu girin:

```
nano ~/.config/user-dirs.dirs
```

Bu komut, user-dirs.dirs dosyasını açacaktır. Bu dosya, kullanıcı klasörlerinin yolunu ve adını içerir. Dosyayı düzenlemek için, kaldırmak istediğiniz klasörün satırını bulun ve "#" işaretiyle başlayan bir yorum satırı yapın. Örneğin, "Masaüstü" klasörünü kaldırmak istiyorsanız, satırı aşağıdaki gibi düzenleyin:

```
#XDG_DESKTOP_DIR="$HOME/Masaüstü"
```

Dosyayı kaydetmek ve kapatmak için "Ctrl + X" tuşlarına basın, ardından "Y" tuşuna basın ve Enter tuşuna basın. Değişikliklerin etkili olması için oturumu kapatıp tekrar açın veya aşağıdaki komutu girin:

```
xdg-user-dirs-update
```

Bu adımları takip ederek user-dirs klasörünü Linux sisteminizden kaldırabilirsiniz. Ancak, dikkatli olun ve yanlışlıkla önemli dosyaları silmemeye özen gösterin.

Yeni oluşturulacak kullanıcılarda oluşturulmamasını istiyorsak **/etc/skel/.config/user-dirs.dirs** dosyasını silmeliyiz.

Sistem Dili Değiştirme

Dil kodlarına /usr/share/i18n/locales içerisinden ulaşabilirsiniz.

Karakter kodlamalara /usr/share/i18n/charmaps içinden ulaşabilirsiniz.

Sistem dilini ayarlamak için öncelikle /etc/locale.gen dosyamızı aşağıdaki gibi düzenleyelim.:

```
tr_TR.UTF-8 UTF-8
```

Not: En altta boş bir satır bulunmalıdır.

Ardından /lib64/locale dizini yoksa oluşturalım.:

```
mkdir -p /lib64/locale/
```

Şimdi de çevresel değişkenlerimizi ayarlamak için /etc/profile.d/locale.sh dosyamızı düzenleyelim.:

```
#!/bin/sh
# Language settings
export LANG="tr_TR.UTF-8"
export LC_ALL="tr_TR.UTF-8"
```

Not: Türkçe büyük küçük harf dönüşümü (i -> İ ve ı -> I) ascii standartına uyumsuz olduğu için LC_ALL kısmını türkçe ayarlamayı önermiyoruz. Bunun yerine C.UTF-8 veya en_US.UTF-8 olarak ayarlayabilirsiniz.

Son olarak locale-gen komutunu çalıştıralım.:

```
locale-gen
```

Eğer /lib64/locale/ dizine okuma iznimiz yoksa verelim.:

```
chmod 755 -R /lib64/locale/
```

Çevresel Değişken Ayarlama Yöntemleri

1. Yöntem

/etc/default/locale dosyasını root olarak bir metin editörü ile açın.

- **Türkçe için :** LANG=tr_TR.UTF-8
- **İngilizce için :** LANG=en_US.UTF-8

Sistemi yeniden başlattığınızda seçtiğiniz dil aktif olacaktır.

2. Yöntem

/etc/profile.d/locale.sh dosyanı oluşturun içeriğini aşağıdaki gibi ayarlayın.

```
# Language settings
export LANG="tr_TR.UTF-8"
export LC_ALL="tr_TR.UTF-8"
```

/etc/profile.d/ dizin erişim iznini 755 yapın.

3.Yöntem

ayarlarını değiştirmek istediğimiz kullanıcı dizinideki **~/.profile** dosyasının içeriğine aşağıdaki kod satırını eklemeliyiz.

```
export LANG="tr_TR.UTF-8"
export LC_ALL="tr_TR.UTF-8"
```

Profile Dosyası

/etc/profile dosyası, Linux sistemlerinde kullanıcıların oturum açtıklarında çalıştırılan bir betik dosyasıdır. Bu dosya, tüm kullanıcılar için ortak kabuk ayarlarını ve ortam değişkenlerini tanımlar.

Kullanıcı oturumu başlatıldığında, **/etc/profile** dosyası sistem genelindeki ayarları yükler ve kullanıcı oturumuna uygular. Bu dosya, kullanıcıların kabuk ortamlarını yapılandırmak ve özelleştirmek için kullanılır.

Örneğin, PATH değişkenini tanımlamak veya diğer kabuk ayarlarını yapılandırmak için /etc/profile dosyası düzenlenebilir. Bu dosya, sistem genelinde tutarlı bir kabuk ortamı sağlamak için önemlidir.

```
/etc/profile
/etc/profile.d/*
/etc/environment
~/.profile, veya ~/.bash_profile veya ~/.login veya ~/.zprofile giriş kabuğunuza bağlı olarak
~/.pam_environment
(yalnızca terminalde çalışan kabuklar için) /etc/bash.bashrc, /etc/zshrc, ~/.bashrc, ~/.zshrc vb.
```

Not: /etc/profile.d/ dizinine root dışında kullanıcılar erişim sağlaması için 755 yapmalısınız.

profile

/etc/profile dosyanının içerisinde aşağıdaki betik olmalıdır. Bu betik **/etc/profile.d** içerisinde betikler varsa tüm kullanıcılar için çalıştırılmasını sağlar.

```
if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
        if [ -r $i ]; then
          . $i
        fi
  done
  unset i
fi
```

Kullanıcı Ekleme

linux sistemlerine kullanıcı eklemek için iki farklı komut bulunur. Bu komutlar adduser ve useradd komutlarıdır.

adduser:

Kullanıcı dostu bir arayüz sunar. Birçok aşamayı bizim adımıza yapar. Temelde useradd komutunu kullanarak yazılan bir script olarak düşünebiliriz.

```
# adduser komutuyla kullanıcı oluşturma
sudo adduser yeni_kullanici
```

useradd:

Kullanıcıdan tüm parametreleri girmesini ister. Bu sebepten çok tercih edilmemektedir. Fakat özel bir şekilde kullanıcı oluşturulmak istenildiğinde tercih edilir. Bu dokümanda kurulum aşamalarında useradd komutu kullanıldı.

```
# useradd komutuyla kullanıcı oluşturma
sudo useradd -m -s /bin/bash yeni_kullanici -d /home/yeni_kullanici
# -s/bin/bash : kullanıcının kullanacağı shell belirtiliyor.
# -d /home/yeni_kullanici : home dizinine oluşturulacak kullanıcı dizini belirtiyoruz
```

cgroup

Cgroup, Linux sistemlerinde kaynak yönetimi için güçlü bir araçtır. Özellikle sunucu ve konteyner ortamlarında, kaynakları etkili bir şekilde yönetmek ve izlemek için vazgeçilmezdir. Yukarıda bahsedilen adımları takip ederek, cgroup ile kaynak yönetimi yapmaya başlayabilirsiniz. Unutmayın, her zaman sistem kaynaklarınızı izlemek ve gerektiğinde ayarlamalar yapmak önemlidir.

Cgroup'un Temel Özellikleri

- **Kaynak Sınırlama:** Cgroup, belirli bir grup süreç için CPU, bellek, disk ve ağ gibi kaynakları sınırlamanıza olanak tanır. Örneğin, bir uygulamanın bellek kullanımını 512 MB ile sınırlamak istiyorsanız, cgroup kullanarak bunu kolayca yapabilirsiniz.
- **Kaynak İzleme:** Cgroup, süreçlerin kaynak kullanımını izlemek için de kullanılabilir. Bu, sistem yöneticilerinin hangi süreçlerin ne kadar kaynak kullandığını görmesine yardımcı olur.
- **Hiyerarşi:** Cgroup, hiyerarşik bir yapıya sahiptir.
- **Güvenlik:** Belirli süreçlerin diğer süreçlerin kaynaklarına erişimini kısıtlayarak güvenliği artırır.

Cgroup Kullanımı

Cgroup kullanmaya başlamak için öncelikle sisteminizde cgroup'un etkin olduğundan emin olmalısınız. Genellikle modern Linux dağıtımlarında cgroup varsayılan olarak aktiftir. Cgroup ile çalışmak için aşağıdaki adımları izleyebilirsiniz:

1. Cgroup Oluşturma

Öncelikle, bir cgroup oluşturmalısınız. Bunun için terminalde aşağıdaki komutu kullanabilirsiniz:

```
sudo mkdir /sys/fs/cgroup/memory/my_cgroup
```

Bu komut, "my_cgroup" adında bir bellek cgroup'u oluşturur.

2. Kaynak Sınırlama

Oluşturduğunuz cgroup'a bellek sınırı eklemek için şu komutu kullanabilirsiniz:

```
echo 512M | sudo tee /sys/fs/cgroup/memory/my_cgroup/memory.limit_in_bytes
```

Bu komut, "my_cgroup" için bellek sınırını 512 MB olarak ayarlar.

3. Süreç Ekleme

Artık bir cgroup oluşturduğunuza göre, bu cgroup'a süreç ekleyebilirsiniz. Bunun için, eklemek istediğiniz sürecin PID'sini öğrenin ve aşağıdaki komutu kullanın:

```
echo <PID> | sudo tee /sys/fs/cgroup/memory/my_cgroup/cgroup.procs
```

Burada <PID> kısmını eklemek istediğiniz sürecin PID'si ile değiştirin.

4. İzleme

Cgroup'un kaynak kullanımını izlemek için aşağıdaki komutu kullanabilirsiniz:

```
cat /sys/fs/cgroup/memory/my_cgroup/memory.usage_in_bytes
```

Bu komut, "my_cgroup" içindeki süreçlerin toplam bellek kullanımını gösterir.

Kaynaklar

- https://tr.wikipedia.org/wiki/Linux - 02/07/2025
- https://www.subrat.info/build-kernel-and-userspace/ - 08/07/2025
- https://medium.com/@chienhaotan/compiling-and-running-a-minimal-kernel-with-busybox-bfc45a991017 - 08/07/2025
- https://stackoverflow.com/questions/64838052/how-to-delete-n-characters-appended-to-ldd-list - 20/06/2025
- https://gist.github.com/bluedragon1221/a58b0e1ed4492b44aa530f4db0ffef85 - 09/07/2025
- https://app.diagrams.net/
- https://www.ubuntubuzz.com/2021/04/how-to-boot-uefi-on-qemu.html - 20/06/2024
- https://wiki.gentoo.org/wiki/OpenRC - 30/06/2025
- https://busybox.net/ - 01/07/2025
- https://busybox.net/about.html - 01/07/2025
- https://sulincix.gitlab.io/distro-kitabi/ - 05/07/2025
- https://gitlab.com/turkman/devel/doc/wiki/ - 05/07/2025
- https://turkman.gitlab.io/devel/doc/wiki/ - 05/07/2025
- https://grok.com/share/c2hhcmQtMw%3D%3D_5c049e44-be57-4652-872f-55def669d917 - 09/07/2025
- https://www.linuxfromscratch.org/lfs/
- https://wiki.archlinux.org/title/GRUB_(T%C3%BCrk%C3%A7e)#UEFI_sistemler - 08/07/2025
- https://tldp.org/HOWTO/html_single/SquashFS-HOWTO/ - 09/07/2025
- https://askubuntu.com/questions/437880/extract-a-squashfs-to-an-existing-directory - 11/07/2025
- https://grok.com/share/c2hhcmQtMw%3D%3D_2ad2ac6b-d067-40b0-9b55-46db0c2c98dc - 10/07/2025
- https://chatgpt.com/

Not: Metin düzenlemelerinde chatgpt kullanılmıştır.

Geliştiricilere Mesajımız

Gnu/Linux, açık kaynaklı bir işletim sistemidir. Kullanıcı ve geliştiriclere bir çok avantajlar sunmaktadır. Bunlar;

- Kodlarına erişilebilir(Açık Kaynak)
- Dağıtılabir
- Değiştirilebilir
- Virüsten etilenme azdır
- Özelleştirilebilir
- Bağımlılığı azaltır
- Güvenlik en önemli şarttır
- Düşük donanımlarda iyi performan verir

Bu özellikleri açısından Gnu/Linux tercih etmek çok avantajlıdır. Geliştirme aşamalarına destek vermek, öğrenmek bize, toplumumuza ve ülkemize sayısız faydalar saylayacaktır.

Lisans Bilgileri

1. **Kaynak Kodlar (Source Code)**:

   Bu dokümandaki kaynak kodların tamamı Free Software Foundation tarafından yayınlanan GNU Genel Kamu Lisansı'nın (GPL) 3. versiyonu ile lisanslıdır.

   Lisansın bir kopyasını şu adresten edinebilirsiniz: https://www.gnu.org/licenses/gpl-3.0.html

İletişim

- https://github.com/kendilinuxunuyap
- https://kendilinuxunuyap.github.io/wm
- kendilinuxunuyap@gmail.com

ISBN: 978-625-90289-0-3

Yayın Yılı: Nisan 2026