Usando Enlightenment DR17 con Slackware

¿Qué es E17?

Enlightenment DR17 (E17) es el tan esperado sucesor de E16, un gestor de ventanas que se distribuyó en Slackware hace 10 años. Se clasifica como un “shell de escritorio”, proporcionando las cosas que necesita para operar su escritorio (o computadora portátil), pero no un conjunto completo de aplicaciones. Esto incluye iniciar aplicaciones, administrar sus ventanas y realizar otras tareas del sistema como suspender, reiniciar, administrar archivos, etc.

E17 no viene solo. En realidad, usa las librerias de Enlightenment Foundation Libraries (EFL) que se puede usar para desarrollar otras aplicaciones como en el mundo Qt / KDE, por ejemplo. Hay (febrero de 2013) 15 bibliotecas en EFL y muchos módulos externos o aplicaciones que se pueden usar con E17.

Installing E17

Paquetes preconstruidos

SlackE17

La forma más fácil y rápida de ejecutar E17 en Slackware es instalar los paquetes E17 desde Proyecto SlackE17. Solo necesita tener la última versión de Slackware, Slackware64 o ArmedSlack.

Solo descargar el tarball correspondiente a su arquitectura (i486, x86_64 o ARM), descomprímalo y lea el documento. Para instalar los paquetes, tiene varias opciones:

- Instalar todo usando:

```
# upgradepkg --install-new e17/*.txz
```

- Use el script provisto para instalar solo E17 y sus dependencias usando:

```
# ./slacke17-install.sh std
```

- Use el script provisto para instalar todo usando:

```
# ./slacke17-install.sh max
```

- O simplemente puedes:

```
# cd e17 && pkgtool
```

Compilación

Primero debe instalar las dependencias necesarias para compilar Enlightenment. Estos se pueden encontrar en el sitio web de SlackBuilds o con sbopkg:

- check
- lua
- luajit
- bullet

Una vez que tenga esos paquetes instalados, podemos proceder a crear un directorio de compilación y comenzar con las cosas divertidas.

```
# mkdir -p /build/e22/core

# cd /build/e22/core

# git clone https://git.enlightenment.org/core/efl.git

# git clone https://git.enlightenment.org/core/enlightenment.git

# cd /build/e22/core/efl

# ./autogen.sh --prefix=/opt/e22 --disable-systemd --disable-avahi
```

Cree un enlace simbólico para dbus (el instalador le indica que haga esto, por lo tanto, compruebe la salida si se ha actualizado solo para asegurarse).

```
# ln -s /opt/e22/share/dbus-1/services/org.enlightenment.Ethumb.service
/usr/share/dbus-1/services/org.enlightenment.Ethumb.service

# make && make install
```

Ahora necesitamos agregar el directorio de la biblioteca de Enlightenment al sistema para que sepa dónde buscar las bibliotecas.

```
# echo "/opt/e22/lib" >> /etc/ld.so.conf
```

Cargue la nueva configuración.

```
# ldconfig
```

Se debe establecer una variable ambiental para PKGCONFIG, esto solo se necesita al construir y no necesita ser permanente.

```
# export
PKG_CONFIG_PATH=/opt/e22/lib/pkgconfig:/usr/lib64/pkgconfig:/usr/share/pkgco
nfig
```

Agregue el contenedor de iluminación a PATH para la compilación, también haga que persista

después del reinicio.

```
# export PATH=/opt/e22/bin:$PATH

# vi /etc/profile
```

Agregue “/opt/e22/bin” a su RUTA.

Comience a compilar Enlightenment.

```
# cd /build/e22/core/enlightenment

# meson --prefix=/opt/e22 -Dsystemd=false -Davahi=false . build

# meson configure build

# ninja -C build install
```

(Opcional, compialr Terminology. Terminology es la terminal predeterminada en Enlightenment).

```
# mkdir /build/e22/apps

# cd /build/e22/apps

# git clone https://git.enlightenment.org/apps/terminology.git

# cd terminology

# meson --prefix=/opt/e22 . build

# meson configure build

# ninja -C build install
```

Para que su administrador de sesiones encuentre Enlightenment, debe crear un archivo .desktop en el directorio correcto.

```
# vi /usr/share/xsession/enlightenment.desktop
```

Debería verse así (consulte los archivos .desktop ya existentes para ver ejemplos de otros idiomas y agregue líneas relevantes para su idioma):

```
[Desktop Entry]

Encoding=UTF-8

Type=XSession

Exec=/opt/e22/bin/enlightenment_start
```

```
TryExec=/opt/e22/bin/enlightenment_start

DesktopNames=Enlightenment

Name=Enlightenment

Comment[en_GB]=Enlightenment by Enlightenment

Comment[sv]=Enlightenment av Enlightenment

X-KDE-PluginInfo-Version=0.22.9
```

Si quieres comenzar la enlightenment con startx (xinit): /etc/X11/xinit/xinitrc.e22

```
#!/bin/sh

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/etc/X11/xinit/.Xresources
sysmodmap=/etc/X11/xinit/.Xmodmap

# merge in defaults and keymaps

[ -f $sysresources ] && xrdb -merge $sysresources
[ -f $sysmodmap ] && xmodmap $sysmodmap
[ -f $userresources ] && xrdb -merge $userresources
[ -f $usermodmap ] && xmodmap $usermodmap

if [ -z "$DESKTOP_SESSION" -a -x /usr/bin/ck-launch-session ]; then
  export DESKTOP_SESSION="e22"
  exec ck-launch-session dbus-launch --exit-with-session enlightenment_start
else
  export DESKTOP_SESSION="e22"
  exec enlightenment_start
fi
```

SlackE17

SlackE17 proporciona todo el código fuente y SlackBuilds en un tarball para que pueda reconstruir todo fácilmente. Puede compilar los paquetes uno por uno, pero es más eficiente usar el script BUILD / slacke17-build.sh. Puede pasar algunas opciones al script:

- Para obtener archivo de depuración:

```
# DEBUG=1 ./slacke17-build.sh
```

- Para optimizar para su maquina:

```
# ECFLAGS="-march=native -mtune=native" ./slacke17-build.sh
```

SBo

SBo proporciona SlackBuilds para EFL y E17, pero tendrá que descargar todo el código fuente usted mismo y construir los paquetes uno por uno.

Usando E17

Necesita actualizar su .xinitrc usando xwmconfig y eligiendo xinitrc.e17. La mayor parte de la configuración se puede hacer con el menú del botón derecho. Si desea administrar quién puede usar el menú del sistema, debe echar un vistazo a /etc/enlightenment/sysactions.conf.

Sources

howtos, e17, author ngc891

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/es:howtos:window_managers:installing_e17

Last update: **2019/08/08 18:35 (UTC)**