

## SlackBuild script –ek készítése

Ez a leírás a [http://www.slackwiki.org/Writing\\_A\\_SlackBuild\\_Script](http://www.slackwiki.org/Writing_A_SlackBuild_Script) oldalon található leírásnak egy kivonatos fordítása, bár néhány helyen ahol szükségesnek találtam kiegészítettem a leírást. A célom nem *műfordítás* készítése volt, hanem az, hogy az angol nyelvű leírásban szereplő információk a magyar felhasználók számára is elérhetőek legyenek saját anyanyelvükön.

Készítette: Polnai Krisztián  
2010.06.30

A leírás feltételezi a Slackware csomagok felépítésének és a Slackware csomagkezelésének ismeretét.

A következő példában a latex2html programhoz fogunk csomagot készíteni.

### Előkészületek

Létrehozzuk a <build\_environment>/latex2html/ nevű könyvtárat, majd letöltjük a latex2html honlapjáról a forráskódot:

```
$ cd <build_environment>
$ mkdir latex2html
$ cd latex2html
$ wget http://saftsack.fs.uni-bayreuth.de/~latex2ht/current/latex2html-2002-2-1.tar.gz
```

Létrehozuk az üres slack-desc és az SBs file –t, a tartalmat később írjuk bele:

```
$ touch latex2html.SlackBuild
$ touch slack-desc
```

Kicsomagoljuk a forrás file –t, mert meg kell majd határoznunk, hogy milyen fordítási opciókra lesz szükségünk:

```
$ tar -xzf latex2html-2002-2-1.tar.gz || exit 1
```

### A SBs elkészítése

#### Kezdő lépések

Nyisd meg a latex2html.SlackBuild nevű file –t a kedvenc szövegszerkesztőben.  
Első lépésként be kell állítani a shell interpreter –t:

```
#!/bin/sh -e
```

Az –e kapcsoló megadásával a shell ki fog lépni bármilyen hiba esetén.  
Lehetőség van licence megadására a következő formában:

```
#<A neved> revision date yyyy/mm/dd
```

A következőkben beállítunk néhány változót. A mi esetünkben a „CWD” változó a <build\_environment>/latex2html/ lesz. Majd megnézzük, hogy a TMP változó be van –e állítva, ha nem, akkor beállítjuk a /tmp –re:

```
#Változók beállítása:
CWD=$(pwd)
if [ "$TMP" = "" ]; then
    TMP=/tmp
fi
```

Van, aki a /tmp –ben megad egy alkönyvtárat (pl.:/tmp/build), de ez nem kötelező.

```
# Verzió információ ami megjelenik az alkalmazás file nevében.
VERSION=2002-2-1

# A "-" karakter nem megengedett a VERSION string -ben.
# A PKG_VERSION -t másra is beállíthatod, mint ami a VERSION string -ben van.
PKG_VERSION=2002.2.1      # A verzió infó, ami megjelenik a csomagnévben.

ARCH=${ARCH:-i486}      # Az architektúra, amire a csomagot fordítod.

#Az első szám a „build number”: hányszor lett a csomag elkészítve.
#A második szó a készítő rövidített neve, tipikusan az első 3 betű.
BUILD=${BUILD:-1_rlw}

# Az alkalmazás neve:
APP=latex2html

# A csomag neve:
PKG=$TMP/package-$APP
```

A következőkben beállítjuk a SLKCFLAGS változót, ami CFLAGS –ként és CXXFLAGS –ként fog működni.

```
if [ "$ARCH" = "i486" ]; then
    SLKCFLAGS="-O2 -march=i486 -mtune=i686"
elif [ "$ARCH" = "x86_64" ]; then
    SLKCFLAGS="-O2 -fPIC"
fi
```

### **A forráskód kicsomagolása**

```
# Ha az install könyvtár létezik, akkor kitöröljük és újra létrehozuk
rm -rf $PKG
mkdir -p $TMP $PKG
rm -rf $TMP/$APP-$VERSION

# Belépünk a TMP -ben megadott könyvtárba.
cd $TMP || exit 1

# Forráskód kicsomagolása
# Megjegyzés: tar.bz2 csomagolás esetén, tar -jxvf beállítást kell használni
tar -zxvf $CWD/$APP-$VERSION.tar.gz || exit 1

# Belépünk az alkalmazás könyvtárába
cd $APP-$VERSION || exit 1
```

```
# Megváltoztatjuk a file -ok tulajdonosát és a hozzáférési jogokat.
# (Erre nincs mindig szükség.)
chown -R root:root .
chmod -R u+w,go+r-w,a-s .
```

## **Konfigurálás és fordítás**

```
# configure beállításainak megadása
# C++ -ban írt alkalmazás esetén egy újabb sort hozzá kell adni a CXXFLAGS
#számára
CFLAGS="$SLKCFLAGS" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--with-perl=/usr/bin/perl \
--enable-eps \
--enable-gif \
--enable-png \
--build=$ARCH-slackware-linux \
--host=$ARCH-slackware-linux

# Forráskód fordítása és hiba esetén kilépés
make || exit

# Mindent felinstallálni a csomag könyvtárába, hiba esetén kilépés
make install DESTDIR=$PKG || exit
```

A következő három configure opciót mindig megadom:

- --prefix=/usr
- --sysconfdir=/etc
- --localstatedir=/var

A konfigurációs file -ok a /etc -be mennek, az állapot („state”) file -ok, mint például a log file -ok a /var könyvtárba kerülnek, a többi dolog pedig a /usr -be.

Az összes konfigurációs lehetőséget, megtekintheted, ha belépsz a könyvtárba, ahová kicsomagoltad a forráskódot és kiadod a következő parancsot:

```
./configure --help
```

A DESTDIR változó nagyon fontos, mert ez határozza meg, hogy a file -ok hova legyenek installálva. Ennek mindig meg kell egyeznie a csomag könyvtárral (\$PKG). Sajnos nem minden alkalmazás Makefile -ja támogatja a DESTDIR használatát, ez letehethető a következő paranccsal:

```
grep DESTDIR Makefile*
```

Ha kapsz valamilyen kimenetet, akkor a DESTDIR változó használata támogatott. Ellenkező esetben a Makefile nem támogatja a DESTDIR használatát.

Megjegyzés: Mindig csináld végig az installációt NORMÁL USER –ként:

```
./configure && make && make install DESTDIR=/valamilyen_könyvtár
```

módon, mert van néhány alkalmazás, ami „vicces” dolgokat művelhet, például a apcupsd alkalmazás patch –eli a /etc/rc.d/rc.6 scriptet és NORMÁL USER –ként ilyenkor „permission denied” üzenetet kapsz. (Észre fogod venni, és a módosítás nem fog végbemenni.)

### **Dokumentáció installálása**

```
# Elkészítjük a doksinak a könyvtárat.
mkdir -p $PKG/usr/doc/$APP-$VERSION

# Dokumentáció átmásolása a célkönyvtárba és a jogosultságok beállítása
cp -a BUGS Changes FAQ INSTALL LICENSE MANIFEST README TODO docs/ \
  $PKG/usr/doc/$APP-$VERSION
find $PKG/usr/doc/$APP-$VERSION -type f -exec chmod 644 {} \;

# Én be szoktam másolni az SBs -et is.
cat $CWD/$APP.SlackBuild > $PKG/usr/doc/$APP-$VERSION/$APP.SlackBuild
```

Győződj meg arról, hogy az adott alkalmazás nem tartalmaz –e más könyvtárakban is dokumentációs file –okat.

### **Végső beállítások**

```
# A ./install könyvtár létrehozása és a slack-desc belemásolása:
mkdir -p $PKG/install
cat $CWD/slack-desc > $PKG/install/slack-desc
```

Megjegyzés: Bizonyos esetekben előfordulhat, hogy a csomag installálása után végre kell hajtani néhány parancsot. Erre szolgál a doinst.sh file.

```
# Ha létezik a doinst.sh, akkor hozzáadjuk a csomaghoz
if [ -e $CWD/doinst.sh.gz ]; then
  zcat $CWD/doinst.sh.gz > $PKG/install/doinst.sh
fi
```

A helytakarékoság érdekében:

```
# Library -k és bináris file -ok strip -elése
( cd $PKG
  find . | xargs file | grep "executable" | grep ELF | cut -f 1 -d : | \
    xargs strip --strip-unneeded 2> /dev/null
  find . | xargs file | grep "shared object" | grep ELF | cut -f 1 -d : | \
    xargs strip --strip-unneeded 2> /dev/null
)
```

```

# Man file -ok becsomagolása, ha léteznek
if [ -d $PKG/usr/man ]; then
  ( cd $PKG/usr/man
    find . -type f -exec gzip -9 {} \;
    for i in $(find . -type l) ; do ln -s $(readlink $i).gz $i.gz ; rm $i ; \
      done
    )
fi

# Ha az info oldalak léteznek, akkor betömörítjük őket és eltávolítjuk a dir
# file-t:
if [ -d $PKG/usr/info ]; then
  gzip -9 $PKG/usr/info/*.info
  rm -f $PKG/usr/info/dir
fi

```

### **A csomag elkészítése**

```

# A csomag elkészítése:
cd $PKG
/sbin/makepkg -l y -c n $TMP/$APP-$PKG_VERSION-$ARCH-$BUILD.tgz

```

### **További észrevételek**

#### **A DESTDIR nem támogatott**

Van néhány alkalmazás, amely Makefile -ja nem támogatja a DESTDIR opciót a make install -nál. Néhány alkalmazás esetén a DESTDIR opciónak más a neve. Például néhány Qt alkalmazás esetén az INSTALL\_ROOT változót használják ugyan arra a célra. Ha értesz a Makefile -okhoz, akkor valószínűleg megéri időt szánnod arra, hogy megnézd, hogy az „install rule” -nál milyen akciók hajtodnak végre.

Előfordulhat, hogy egyáltalán nincs a DESTDIR -nek megfelelő változó. Ilyenkor a legjobb, amit tehetsz, hogy írsz egy patch -et a Makefile.in -re, vagy esetleg meg kell kérni a fejlesztőt, hogy a következő kiadásokba építse be ezt a funkciót.

De addig is itt van néhány példa:

#### **Példa1:**

A következő beállításokkal minden file a \$PKG könyvtárba fog kerülni:

```

./configure --prefix=$PKG/usr \
  --sysconfdir=$PKG/etc \
  --localstatedir=$PKG/var

```

Ezután a file -okat át kell mozgatni a megfelelő helyre.

#### **Példa2:**

Ha a DESDIR macro nem támogatott, akkor célszerű lehet a prefix macro megváltoztatása:

```

make PREFIX=$PKG/usr install

```

Ezzel a beállítással az installációs könyvtár megváltoztatható. (A Makefile -ban a \$(prefix) változó íródik át.)

Megjegyzések:

1. A configure beállításainál a következő továbbra is használható: `./configure --prefix=/usr`, de a tényleges installációs könyvtár más lesz.
2. Vannak alkalmazások, ahol a „PREFIX=” helyet a „prefix=” makrót használják.

### **A forráskód patch –elése**

Először is a patch file –t le kell tölteni és be kell rakni abba a könyvtárba, ahol az SBs és a többi szükséges file található:

```
$ wget http://someapplication.org/files/patches/bigsecuritypatch.diff
```

A következő lépés nem szükséges, de helyet takarítunk meg vele:

```
$ gzip -9 bigsecuritypatch.diff
```

Ennek az eredménye egy bigsecuritypatch.diff.gz nevű file lesz.

Az SBs file –t módosítsd úgy, hogy a forráskód kicsomagolása után, de még a `./configure` elé tedd be a következő sort:

```
zcat $CWD/bigsecuritypatch.diff.gz | patch -pX || exit
```

Az „X” helyére egy szám kerül, az úgy nevezett „patch level”. Ez alkalmazás függő; általában 1 vagy 0. A `-pX` azt határozza meg, hogy a file nevek, hogyan lesznek kezelve a patch –ben. Azaz a `-p` után megadott szám megadja, hogy hány könyvtárat hagyjon el a file elérési útvonalából.

Például, ha a patch –ben lévő file –név: `/u/howard/src/blurfl/blurfl.c`, akkor a `-pX` kapcsoló a következő képen fogja átalakítani a file nevet:

- `-p` esetén: `blurfl/blurfl.c`
- `-p0` esetén: `/u/howard/src/blurfl/blurfl.c`
- `-p1` esetén: `u/howard/src/blurfl/blurfl.c`
- `-p4` esetén: `blurfl.c`