chromeupdate.nw – automatically update Chrome

Edward McGuire

April 2, 2024

1 copyright notice

Contents

2 chromeupdate.sh – check and update Chrome

`chromeupdate.sh` is a script written in Bourne-family command language.

The steps taken are:

1. Get the latest Google package. Stop if
   - the request failed, or
   - we had the latest Google package already.
2. Convert to a Slackware package using the tool provided.
3. Perform a Slackware package install.

4 ⟨*chromeupdate.sh* 4⟩≡

```
#! /bin/sh
# This file is part of the chromeupdate package.
# Compiled from chromeupdate.nw using Noweb by Norman Ramsey.
⟨chromeupdate.sh elevate 5⟩
⟨chromeupdate.sh get 6c⟩
⟨chromeupdate.sh convert 7⟩
⟨chromeupdate.sh install 8⟩
echo "$0: success: ⟨chrome package 6a⟩ installed"
```

This code is written to file `chromeupdate.sh`.

2.1 Elevate

The update should be run with elevated (root) privilege so it can run `upgradepkg`.

5 ⟨*chromeupdate.sh elevate* 5⟩≡ (4)

```
test ‘ id -u ‘ = 0 ||
{
        echo "$0: must be su"
        exit 1
}
```

2.2 Get

The Google package we want is Debian stable 64-bit.

6a ⟨*chrome package* 6a⟩≡ (4 6)
```
google-chrome-stable_current_amd64.deb
```

6b ⟨*chrome url* 6b⟩≡ (6c)
```
https://dl.google.com/linux/direct/⟨chrome package 6a⟩
```

Conditionally download the Chrome package using `wget`. If `wget` throws an error, or nothing is downloaded, stop.

6c ⟨*chromeupdate.sh get* 6c⟩≡ (4) 6f▷
```
OLD=‘ date -r⟨chrome package 6a⟩ +%s 2>/dev/null ‘
wget --no-verbose --timestamping ⟨chrome url 6b⟩ ||
{
        echo "$0: fatal: failed to get ⟨chrome package 6a⟩"
        exit $?
}
NEW=‘ date -r⟨chrome package 6a⟩ +%s 2>/dev/null ‘
test "$OLD" = "$NEW" &&
{
        echo "$0: success: ⟨chrome package 6a⟩ did not change"
        exit 0
}
```

The Slackware conversion script we want is the Slackware 15.0 64-bit.

6d ⟨*slackbuild script* 6d⟩≡ (6 7)
```
google-chrome.SlackBuild
```

6e ⟨*slackbuild url* 6e⟩≡ (6f)
```
https://mirrors.slackware.com/slackware/slackware64-15.0/extra/google-chrome/⟨slackbuild script 6d⟩
```

Conditionally download the script using `wget`. If `wget` throws an error, stop.

6f ⟨*chromeupdate.sh get* 6c⟩+≡ (4) ◁6c
```
wget --no-verbose --timestamping ⟨slackbuild url 6e⟩ ||
{
        echo "$0: fatal: failed to get ⟨slackbuild script 6d⟩"
        exit $?
}
```

2.3 Convert

Run the Slackware conversion script as root to make the Slackware package. If it throws an error, stop.

7 ⟨*chromeupdate.sh convert* 7⟩≡ (4)

```
sh ⟨slackbuild script 6d⟩ ||
{
        echo "$0: fatal: ⟨slackbuild script 6d⟩ threw an error"
        exit $?
}
```

2.4 Install

Finally, verify there is one good package and install it. Afterward, delete the package. This avoids setting up the next run for failure.

Better would be, instead of finding one good package, find the latest package.

8 ⟨*chromeupdate.sh install* 8⟩≡ (4)

```
cd /tmp ||
{
        echo "$0: fatal: no /tmp"
        exit $?
}
test ‘ ls -d google-chrome*.txz | wc -w ‘ != 1 &&
{
        echo "$0: fatal: too many google-chrome packages in /tmp"
        exit 1
}
upgradepkg --install-new google-chrome*.txz
rm google-chrome*.txz
```

3 Secondary swapfile

The conversion script eats memory. Example of creating a temporary second swapfile:

```
# swapon --show
NAME     TYPE      SIZE   USED PRIO
/dev/sdb partition 512M 105.4M   -2
# df -BM
Filesystem     1M-blocks   Used Available Use% Mounted on
/dev/root         24603M 17525M     5809M  76% /
...
# dd if=/dev/zero of=/swaptemp bs=1M count=2048
# df -BM
Filesystem     1M-blocks   Used Available Use% Mounted on
/dev/root         24603M 19573M     3761M  84% /
...
# chown root.root /swaptemp
# chmod 0600 /swaptemp
# mkswap /swaptemp
# swapon /swaptemp
# swapon --show
NAME      TYPE      SIZE   USED PRIO
/dev/sdb  partition 512M 104.5M   -2
/swaptemp file        2G     0B   -3
# # do what needs a big swap
# swapoff /swaptemp
# rm /swaptemp
```

The script ought to check for ”enough” swap.

4 Makefile

10 ⟨*makefile* 10⟩≡

```
usage :: ; @echo 'usage: make { all | install | commit | clean }'
all ::
install :: all
commit :: ; git commit -av -uno
clean :: ; rm -f *~ .*~
all :: chromeupdate chromeupdate.pdf makefile
chromeupdate : chromeupdate.sh
clean :: ; rm -f chromeupdate
chromeupdate.sh : chromeupdate.sentinel ;
clean :: ; rm -f chromeupdate.sh
chromeupdate.pdf : chromeupdate.tex ; latexmk -pdf $<
install :: /var/www/metaed.com/root/papers/chromeupdate.pdf
/var/www/metaed.com/root/papers/chromeupdate.pdf : chromeupdate.pdf ; cp $< $@
clean :: ; latexmk -C chromeupdate.tex
chromeupdate.tex : chromeupdate.sentinel ;
clean :: ; rm -f chromeupdate.tex
makefile : chromeupdate.sentinel ;
clean :: ; rm -f makefile
chromeupdate.sentinel : chromeupdate.nw ; noweb $< ; touch $@
clean :: ; rm -f chromeupdate.sentinel
```

This code is written to file `makefile`.