Umeå University
Department of Applied Physics and Electronics

Linux as Development Environment 7.5 ECTS
5EL142 HT-16

Assignment 2

Submitted 2016-08-20
Author: Lorenz Gerber, 20161202-2033 (lorenzottogerber@gmail.com)
Instructor: Sven Rönnbäck

1 Describing Basic Linux Commands

man *keyword* display manual pages for a specific command

man -k *expression* display the short descriptions that contain a regex match to *expression*

man -f *keyword* display the short descriptions that contain the *keyword*

info *topic* display info document

info -k *string* search for *string* in all manuals

info -h show help page about info

cp *source dest* copy a file or directory

cp -r *source dest* copy recursively

cp -n *source dest* does not overwrite at dest

mv *source dest* move a file or a directory

mv -n *source dest* no-clobber, don't overwrite

mv -i *source dest* ask interactive before overwrite

mkdir *name* make a directory

mkdir -m *name* set file mode using chmod syntax

rmdir *directory* remove empty directory

rm *file* remove file

rm -r *file/directory* recursively remove *file/directory*

rm -i *file* ask interactively before removing

find . *options startpoint expression* command to search in the file system

find . *-depth 3* an expression global option that defines the depth to search in directories

find . *-type f* an expression test option to find normal files

find . *-exec command* an expression action option to execute *command* on each found file

cd *directory* shell built-in command to change directory

cd */directory* a trailing slash defines an absolut path

pwd diplay present working directory

df reports free disk space

df -h human readable

df -T print type of disk

ps show processes

ps -ax show all process system wide

ps -ax –forest show all processes system wide in a tree format

du . disk usage calculating from current directory

du . *-h* human readable sizes

du . *-d 2* directories deeper than 2 are summarized

tar command to store or extract files from a tar tape/disk archieve

tar *-xvzf name* extract the compressed *name*.tar.gz archieve to the current directory

tar *-cvzf* name files create *name*.tar.gz from *files*

seq 10 generate sequence of numbers

seq 5 10 sequence from 5 to 10, default increment by 1

seq *-s ' '* 5 .5 10 sequence from 5 to 10, increment by 0.5, separate numbers by a whitespace, default is \n

whoami display the name of current user

users user names of currently logged on users at present host

who show information about currently logged in users

who -a show all available information about currently logged in users

who -b show last time of system boot

whereis *command* find binary, source and man pages for a specified command

whereis -b *command* find path to binaries

whereis -s *command* find path to sources

cat *file ...* concatenates files and writes to std out

cat -A *file ...* will write non-printing command characters

cat -n *file ...* number all lines

tee *file ...* writes stdandard in to standard out and file(s)

tee -a *file ...*

more *file* pager for text files

more -10 *file* number of lines to show on each page (10 in this case)

more -s *file* combine multiple blank lines into one

less *file* advanced pager for text files

less -N *file* print line numbers

less -E *file* exit when reaching EOF

uniq *file* matches repeated lines to either report or discard

uniq -c *file* show count of row repetitions

uniq -i *file* ignore case for comparison

tail *file* display last 10 lines of a file

tail -n20 *file* show the last 20 lines of a file

tail -c1024 *file* show the last 1024 bytes of a file

echo *text* display a line of text

echo -n *text* no trailing newline

which *command* find the path to a command

wget *URL* network file downloader

wget -nd *URL* do not create recursive directory structure

wget -b *URL* go to background after download start

cut *file* print select parts of lines (characters, fields) from a file or stdin

cut -c1-4*file* cut and show from each line character 1 to 4 of file

cut -d- -f4-6 *file* cut and display field 4 to 6 when '-' is the delimiter in *file*

grep *pattern file* print lines that match a pattern of interest

grep -F *pattern file* fixed string pattern (not regex)

grep -i *pattern file* ignore case

sort *file ...* sort lines of *file(s)*

sort -d *file ...* dictonary order

sort -b *file ...* ignore leading blanks

wc *file* lines, word and chars count

wc -c *file* chars count

wc -L *file* max line length

2 User Access

The 'root' user has always access.

3 Users and Groups

1. Users *aandersson, ppettersson, lpersson* were created with **adduser**
2. groups *datagroup, admingroup, marketgroup* were created with **groupadd**
3. subdirectories *data, admin, market* were created with **mkdir**
4. directories permissions were set to 770 by **chmod**
5. directories ownership were set to root by **chown**

6. groups were assigned to the respective directories using **chown**
7. users were assigned to the requested groups using **gpasswd -a *user group***

Below follows a screencopy from the directories:

```
drwxrwx--- 2 root    admingroup  4,0K aug 26 23:34 admin
drwxrwx--- 2 root    datagroup   4,0K aug 27 11:25 data
drwxrwx--- 2 root    marketgroup 4,0K aug 26 23:34 market
```

Screen copy of */etc/passwd*:

```
aandersson:x:1001:1001:Adam Andersson,,,:/home/aandersson:/bin/bash
lpersson:x:1002:1002:Lisa Persson,,,:/home/lpersson:/bin/bash
ppettersson:x:1003:1003:Peter Pettersson,,,:/home/ppettersson:/bin/bash
```

Screen copy of */etc/group*:

```
aandersson:x:1001:
lpersson:x:1002:
ppettersson:x:1003:
datagroup:x:1004:ppettersson,aandersson
admingroup:x:1005:lpersson,aandersson
marketgroup:x:1006:ppettersson,lpersson
```

4 Filesystem

/boot contains files to boot the system up

/etc system wide configuration files

/sbin system administration binary exectuables

/bin binary executables for all users

/usr user binary executables, source, doc, and libraries for more complex programs

/var variable files, such as logs, lock files, mails, print queues

/dev device files such as terminals, usb's and other interfaces

/home user home directory

5 Pipes

1. **find . +1M | sort**
2. **sort adress.txt | uniq | wc -l**
3. **who | cut -d' ' -f1 | uniq | sort**
4. **seq 1 10 | tee test1 test2**

6 streams

ls / < lsoutput.txt — redirects the standard out from list directory of root (/) into the file lsoutput.txt. No screenoutput.

ls /hh > lsoutput2.txt — redirects the standard out from list directory of /hh into the file lsoutput2.txt. As there doesn't exist such a directory, lsoutput2.txt is empty while an error message is displayed on the screen.

ls / 2> lserror.txt — Standard error of listing the root directory is redirected into the file lserror.txt. As root exists, no error message is produced hence lserror.txt is an empty file. The directory listing is displayed on the screen.

ls /hh 2> lserror2.txt — Standard error of listing /hh is redirected into the file lserror2.txt. Hence lserror2.txt contains the error message *ls: cannot access '/hh': No such file or directory*.

By default STDIN reads from the keyboard. Alternatively but also by default can a program use the stream from a pipe operator | as STDIN. The control character for STDIN is '<'. It can be used to read STDIN for example from a file. STDOUT is by default to the terminal. The control character '>' is used to divert STDOUT to a file. The file is in this case always new created. The control character sequence '»' can be used to append to an existing file. The pipe operator '|' can be used to divert STDOUT to another command/program.