Linux - a quick Introduction

Caroline Sporleder & Ines Rehbein

WS 09/10

What is a shell?

- Interface between the user and the system
- Command language interpreter that executes commands read from the standard input device (keyboard) or from a file
- Several shells available: sh, bash, csh, tcsh, ksh, ...
- More info:
  `http://www.freeos.com/guides/lsst/ch01sec07.html`
- Find out which shell is the default on your system:

  echo $SHELL

  e.g. /bin/bash   Bourne shell

SSH

- Secure SHell: remote login program
- allows you to log on to a remote machine
- secure, encrypted communication
- more info: man ssh, Linux-Intro 10.4.4 (`tldp.org/LDP/intro-linux/intro-linux.pdf`)

Login/logout

ssh -l user hostname	login `user` onto comuter `hostname`

e.g.:

```
ssh -l rehbein login.coli.uni-saarland.de
ssh -l csporled forbin.coli.uni-saarland.de
```

- for internal use (CIP pool): login/forbin/other server name
- for connecting from outside (from home) whole path needed
  ⇒ server.*coli.uni-saarland.de*

Files and Directories

cd *directory*	change to directory *directory*
pwd	show absolute path of current directory
ls	show all files in current directory
ls -a *directory*	show all files in directory *directory* (including filenames starting with .)
ls -l	show all files in current directory (detailed file information)
cat *file*	print content of file *file* to STDOUT
more *file*	page through file *file* one screenfull at a time
less *file*	like *more*, but more user friendly
cat *f1 f2 >f3*	write content of file *f1* and *f2* to file *f3*
cat *f1 >>f2*	append content of file *f1* to the end of file *f2*
cp *f1 f2*	copy file1 to file2 (rename f1 as f2)
cp *f1 f2 directory*	copy files *f1* and *f2* into directory *directory*
mv *f1 f2*	rename file *f1* as file *f2*
mv *f1 dir*	move file *f1* to directory *dir*
mkdir *directory*	make (create) directory *directory*
rm *file*	remove file *file*
rm -r *dir*	remove directory *dir* and all files/directory in *dir*

Text Editors

- Text editors:
  - gedit (has a GUI), emacs, xemacs, vi, pico, nano, ...

Joe's Own Editor

joe *file*	start joe and open *file*
Strg+Y	delete line
Strg+W	delete all characters up to the end of the word
Strg+KH	show help/quit
Strg+A	move cursor to beginning of line
Strg+E	move cursor to end of line
Strg+X	move cursor to beginning of next word
Strg+C	cancel
Strg+KX	save file and quit

File System Permissions

- File system permissions under UNIX/Linux:
  - files/directories are owned by a user who belongs to a specific group
  - permission to read/write/execute can be defined independently for specific users and groups
  - to start/delete a file you need write/execute permission for the directory
  - to list the content of a directory (ls), you need the permission to read/execute

	user	**group**	**other**	
-	r w x	r w -	r - x	**file**
d	r w x	r w x	r - x	**directory**

File System Permissions

ls -l *file*	show detailed file info for file *file*
ll	same as ls -l
chmod u+x *file*	set permission to execute *file* for user (means user is allowed to execute file)
chmod o+r *file*	set reading permission for other (means file can be read by all users)
chmod g+w *file*	set writing permission for group
chmod 754 *file*	user is allowed to read/write/execute *file* group is allowed to read/execute *file* other is allowed to read *file*
chown *user file*	change owner of *file* to *user* e.g. chown rehbein *file* $\Rightarrow$ *file* is owned by rehbein now
chgrp *group file*	change group of *file* to *group*

Information

who	show who is currently logged in to the system
whoami	show user id
whereis *command*	find binaries, source code or man pages for a specific command (e.g. whereis ruby)
which *command*	show path for a (shell) command
locate *pattern*	find all files in a database which match the *pattern*
man *command*	show manpages (manual) for a command
command - -help	show help for *command*
command -h	same as *command* - -help

Show/Print

echo text	print text to STDOUT
echo text > *file*	write text to file *file*
echo text >> *file*	append text to file *file*
head *file*	print first 10 lines of file *file* to STDOUT
head -100 *file*	print first 100 lines of file *file* to STDOUT
tail *file*	print last 10 lines of file *file* to STDOUT
tail -50 *file*	print last 50 lines of file *file* to STDOUT
cat *file*	print content of *file* to STDOUT

Sort

tac *file*	print text lines of file in reverse order
rev *file*	print characters of file in reverse order
nl *file*	print *file* with line numbers
sort *file*	sort text lines in *file* alphabetically
sort -r *file*	reverse sort
sort -n *file*	sort numerically
sort -u *file*	sort and remove double entries (uniq sort)
cat *file* | sort | uniq -c	show *file*, sort, delete double entries, show frequency of each entry
cat *file* | xargs -n1	show file content, one word per line
cat *file* | xargs -n3	show file, 3 words per line

xargs has problems with ', you can use *tr* instead:
tr ' ' '[RET]
' <*file*

Count and Compare

wc -l *file*	count number of lines in file
wc -w *file*	number of words
wc -c *file*	number of characters
diff *file1 file2*	show all lines in *file1* and *file2* which differ
diff -b *f1 f2*	ignore blank space
diff -i *f1 f2*	ignore upper/lower case

Wildcards

Asterisk *	matches any number of characters, including none
Question mark ?	exactly one character
Square brackets [3-9]	any number between 3 and 9
Square brackets [c-f]	c, d, e or f

ls *.txt	list all file names ending with .txt
mv handout* WS09	move all files starting with "handout" to directory WS09
rm chapter[2-5]	delete chapter2, chapter3, chapter4, chapter5
ls *.[pt][dx][ft]	list all files ending with .pdf and .txt
cp *.htm* *directory*	copy all files ending with .htm, followed by any number of characters, including none, to *directory* e.g. file.htm, file.html
cp *.htm? *directory*	copy all files ending with .htm, followed by exactly one character, to *directory* file.html (but NOT file.htm)

Grep

grep *pattern file*	prints all lines in *file* matching a particular search pattern
grep crisis *file*	prints all lines in *file* containing "crisis"
grep -v crisis *file*	prints all lines in *file* NOT containing "crisis"
grep -o crisis *file*	show only the part of the line that matches "crisis"
grep [Mm]inister[Ii]*n* *file*	show all lines containing the masculine or feminine form of "Minister", and all compound words with "minister"
grep -n *pattern file*	include line numbers
grep - -color *pattern file*	highlightning of search pattern
grep *pattern file* | wc -l	shows number of lines containing search pattern (How often occurs *pattern* in *file*?)

Compressing/Archiving Files

file endings	tar.gz, tgz, zip, gz, bz, rar, bz2, ...
gzip *file*	compress a file using gunzip
gunzip *file.gz*	uncompress a gunzip file
tar -cf *archive.tar f1 f2 f3*	create archive *archive.tar* and add files f1, f2, f3 to archive file
tar -czf *archive.tgz f1 f2 f3*	like *tar -cf*, additionally compress archive using gunzip
tar -xzf *archive.tgz*	decompress archive.tgz and write files to current directory

Job Control

jobs	list all jobs (running processes)
jobs -l	jobs with job id
ps	show all running jobs
kill	send signal to job
kill -l	lists all possible signals
kill -9 PID	send signal SIGKILL (terminate process) to job with id PID
<STRG>-c	terminate current job
<STRG>-z	suspend current job
bg	resume suspended job in the background
bg *job number*	resume job with *job number* in the background
fg *job number*	resume job with *job number* in the foreground
command &	start *command* in the background

Exercise

- Open *gedit* and create a test file, save test file to disk
- Create a new directory *linux_ex1* and move your test file to *linux_ex1*
- Create a new directory *linux_ex2* and copy test file to *linux_ex2*
- Check file permissions: Who owns the file? Who is allowed to read/write/execute it?
- Change file permisssions: allow all users to read the file, but only the owner should be allowed to write/execute it

Exercise

- Copy file *exercise.txt* from /proj/contrib/lpdd to folder *linux_ex2*
- How many lines of text are in the file? How many words? Characters?
- How often occurs *Jesus* in the file?
- Write *exercise.txt* to a new file *exercise.1word* in a one-word-per-line format
- Create a word list (remove double entries) with word frequencies for exercise.1word, write the list to a text file *wordlist.exercise*
- Sort your word list numerically, write the output to a text file *wordlist.exercise.sorted*
- Check for differences between your word list and the file */proj/contrib/lpdd/wordlist*. Are the two files the same? If not, what's the difference?

Useful Links

- Linux Shell Scripting Tutorial v1.05r3
  A Beginner's handbook
  `http://www.freeos.com/guides/lsst`
- The Linux Cookbook: Tips and Techniques for Everyday Use
  `http://dsl.org/cookbook/cookbook_toc.html`
- Analyzing Text
  `http://dsl.org/cookbook/cookbook_16.html`
- Introduction to Linux – A Hands on Guide
  `tldp.org/LDP/intro-linux/intro-linux.pdf`