

LINUX

A QUICK INTRODUCTION



The Unix System

- Unix is case sensitive
- Unix users do not use spaces in filenames
- Most of the work is done from the shell
- Shell acts as interface between user and system
- Bourne shell is the shell used in Unix

```
echo $shell
```

```
/bin/bash
```

Login

- An account, username and password is required to login to a system.
- Remote login requires **Secure SHell** : remote login program
- Allows you to log on to a remote machine
- SSH ensures secure, encrypted communication

Syntax

```
ssh username@hostname
```

or

```
ssh -l username hostname
```

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 screenful 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
rmdir directory	remove directory
rm -r dir	remove directory dir and all files/directory in dir
text-editor-name file	open file file in text editor *Text Editors – vi, gedit (has a GUI), emacs, xemacs, pico, nano ...

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

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

More useful Commands

who	show who is currently logged in to the system
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
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
df	print available and used disk space usage of file system
du	print estimate of file space usage

Wildcards

Asterisk *	matches any number of characters, including none
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
cp *.htm? directory	copy all files ending with .htm, followed by exactly one character, to directory file.html (but NOT file.htm)

Grep

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

Find

<code>find . -name <i>filename</i></code>	find a file by name (case - sensitive)
<code>find <i>/directory</i> -name filename</code>	find file <i>filename</i> under the directory <i>directory</i>
<code>find . -name 'filena'</code>	find file with partial filename
<code>find ~ -iname '*.tex'</code>	find a file by name ignoring the case (~ used for including hidden files)
<code>find ~ -size +50M</code>	find all files greater than 50MB
<code>find ~ -size -100M</code>	find all the files less than 100MB
<code>find . -type f -name "*.py"</code>	find all the files with ending .py
<code>find /home -user math</code>	find file owned by a particular user

File Transfer

While working on a remote system use the following for file transfer:

sftp	sftp -p port username@host
scp	scp username@remotehost.edu: foobar.txt /some/local/directory
rsync	rsync options source destination
ssh with remote display	
Ssh - X or ssh -Y (X11 forwarding)	

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
tar jxf archive.bz2	unpack .bz2 file
tar jcf archive.bz2 f1 f2 f3	create .bz2 archive file
zip zipped.zip f1 f2 f3	create .zip compressed files
unzip zipped.zip	unzip a .zip file

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
pipe () commands	send the output of one program to another program for further processing.
cat filename less	display the contents of file filename one screenful at a time using less
top	show the current processes on the system
./a.out	execute command for current directory

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