

Computer Systems and Architecture

Introduction to UNIX

Stephen Pauwels

University of Antwerp

October 2, 2015

Outline

What is Unix?

Getting started

Streams

Exercises

UNIX

- ▶ Operating system
- ▶ Servers, desktops, laptops
- ▶ Different types
 - ▶ Sun Solaris
 - ▶ FreeBSD
 - ▶ GNU/Linux (Ubuntu, OpenSUSE, Debian, CentOS, ...)
 - ▶ MacOS X

UNIX

- ▶ Three parts:
 - ▶ Kernel
 - ▶ Central component, manages the computer's hardware (CPU, memory, filestore...)
 - ▶ Shell
 - ▶ Interface between user and kernel
 - ▶ Command Line Interpreter (CLI)
 - ▶ Programs

UNIX Shell

- ▶ Interprets commands
- ▶ Commands are themselves programs
- ▶ Filename completion (use the [tab] key)
- ▶ History (use the cursor keys)

Files and processes

- ▶ Everything in UNIX is a file or a process
 - ▶ A process is an executing program (unique PID)
 - ▶ A file is a collection of data
- ▶ Directory structure
 - ▶ Root: /
 - ▶ Home dir: ~/
 - ▶ Current dir: ./
 - ▶ Parent dir: ../
 - ▶ Absolute path: /home/p10/p101234/oefeningen.html
 - ▶ Relative path: ./oefeningen.html

File permissions

▶ Users

- ▶ Unique username
- ▶ Member of one or more groups
- ▶ `/etc/passwd` and `/etc/group`

▶ Files

- ▶ Owner (and associated group)
- ▶ Set of permission flags: `r (+4)`, `w (+2)`, `x (+1)` for owner, group, other
- ▶ Change with `chmod`, `chown`, `chgrp`

▶ Examples:

- ▶ `chmod 755 file` Owner can do everything, group and others can read/execute
- ▶ `chmod 777 file` Everyone can read, write and execute
- ▶ `chmod 600 privatefile` Owner can read/write, others can't do anything
- ▶ `chmod 664 file` Owner and group can read/write, others can only read
- ▶ `chmod +x file` Add execute permissions for everyone

Basic UNIX commando's - browsing

<code>ls</code>	list files and directories
<code>ls -a</code>	list all files and directories
<code>mkdir</code>	make a directory
<code>cd <i>directory</i></code>	change to named directory
<code>cd</code>	change to home directory
<code>cd ~</code>	change to home directory
<code>cd ..</code>	change to parent directory
<code>pwd</code>	display the path of the current directory
<code>find</code>	search through directory tree

Basic UNIX commando's - files

`cp file1 file2`

copy file1 and call it file2

`mv file1 file2`

move or rename file1 to file2

`rm file`

remove a file

`rmdir directory`

remove a directory

`cat file`

display a file

`less file`

display a file a page at a time

`head file`

display the first few lines of a file

`tail file`

display the last few lines of a file

`grep 'keyword' file`

search a file for keywords

`wc file`

count number of metrics in file

`ln -s from to`

make softlink from to

`uniq`

report or filter out repeated lines in a file

`file file`

show the file type of a file

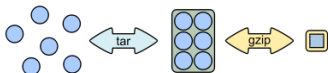
`du file`

show the disk size of a file or directory

Basic UNIX commando's - Varia

<code>man <i>command</i></code>	display manual pages for a command
<code>date</code>	display date and time
<code>who</code>	info on all currently logged on users
<code>whoami</code>	info about yourself
<code>echo <i>hello world!</i></code>	display characters in the terminal
<code>sort text</code>	sort its input
<code>finger</code>	Lookup user info

Archiving



- ▶ Tar: Uncompressed
 - ▶ Create: `tar -cvf tarball.tar files`
 - ▶ Extract: `tar -xvf tarball.tar`
 - ▶ List files: `tar -tf tarball.tgz`
 - ▶ Update files: `tar -uf tarball.tgz files`
- ▶ Gzip: Compression
 - ▶ Create: `gzip tarball.tar`
 - ▶ Extract: `gunzip tarball.tar.gz`
- ▶ tar.gz or tgz?
 - ▶ Tar + compression
 - ▶ Create: `tar -cvzf tarball.tgz files`
 - ▶ Extract: `tar -xvzf tarball.tgz`
 - ▶ List files: `tar -tzf tarball.tgz`

Processes

- ▶ Jobs are connected to terminal which started them
- ▶ Foreground or background
- ▶ Ctrl+C: kill current job

<code>kill [-9] <i>process_id</i></code>	Kill job with pid <i>process_id</i>
<code>pkill <i>process_name</i></code>	Kill job with name <i>process_name</i>
<code><i>command</i> &</code>	Run <i>command</i> in the background
<code>ps [-ef]</code>	Display process info
<code>top</code>	Display process info interactively
<code>jobs</code>	Display user's jobs
<code>fg <i>pid</i></code>	Bring process to the foreground
<code>bg <i>pid</i></code>	Bring process to the background

Basic UNIX commando's - Network

`ssh username@host`

`scp from username@host:to`

`wget http://url/file.jpg`

Login with a remote shell

copy files over network

download files from the web

Standard in- and output

- ▶ Processes write to the standard output, and take their input from the standard input.
- ▶ Keyboard and terminal
- ▶ Standard error (terminal)
- ▶ Redirection is possible
 - ▶ `>`, `>>`, `<`, `2>`

Redirection

- ▶ `ls -alrF > listing.txt` : store ls output in listing.txt
- ▶ `sort < listing.txt` : feed listing.txt to sort program
- ▶ `echo HOI >> listing.txt` : append string HOI to listing.txt
- ▶ `echo hello > /dev/null` : suppress output
- ▶ `who 2> errors.txt` : store errors in file

Pipes

- ▶ Pipes redirect output from one process to the following one
- ▶ Without pipes:

```
who > names.txt  
sort < names.txt
```
- ▶ With pipes:

```
who | sort
```


Exercises

- ▶ `http://msdl.cs.mcgill.ca/people/hv/teaching/ComputerSystemsArchitecture/#CS1`