Computer Systems and Architecture

Introduction to UNIX

Ruben Van den Bossche
Original slides by Bart Meyers

University of Antwerp

Outline

What is Unix?

Getting started

Streams

Exercises

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 Shell

- 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)

Student server

- FreeBSD
- Login using SSH
  - Windows: Putty
  - UNIX: `ssh username@radix.cmi.ua.ac.be`
- Radix
  - Login: `p10xxxx` (roll number)
  - Password: change with command `passwd`
- Webserver
  - Put files in directory `~/public_html`
  - Access through `http://radix.cmi.ua.ac.be/~username/`

Outline

What is Unix?

Getting started

Streams

Exercises

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

Basic UNIX commando’s - browsing

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

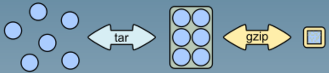
Basic UNIX commando's - files

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
unique	Report or filter out repeated lines in a file

Basic UNIX commando's - Varia

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

Archiving



- Tar
  - Uncompressed
  - Create: `tar -cvf` *`tarball.tar files`*
  - Extract: `tar -xvf` *`tarball.tar`*
- 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`*

Processes

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

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

Basic UNIX commando's - Network

`ssh` *username@host*	Login with a remote shell
`scp` *from username@host:to*	copy files over network
`wget` *http://url/file.jpg*	download files from the web

Outline

What is Unix?

Getting started

Streams

Exercises

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
```

Outline

What is Unix?

Getting started

Streams

Exercises

Exercises

- http://msdl.cs.mcgill.ca/people/hv/teaching/ComputerSystemsArchitecture/#csw2