Computer Systems and Architecture

Introduction to UNIX

Stephen Pauwels

Academiejaar 2016-2017

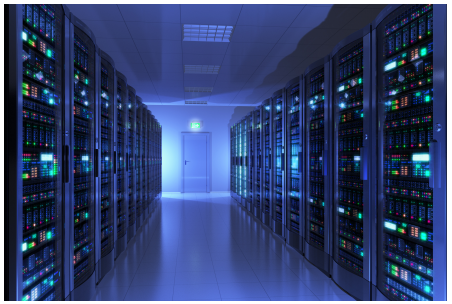
Universiteit Antwerpen

Outline

- What is UNIX?
- Getting started
  - UNIX Shell
  - Files and Processes
  - Basic UNIX Commands
- Input/Output

UNIX

- Operating System (OS)
- Servers, desktops, laptops, tablets




- Different types
  - Sun Solaris
  - FreeBSD
  - GNU/Linux
  - MacOS

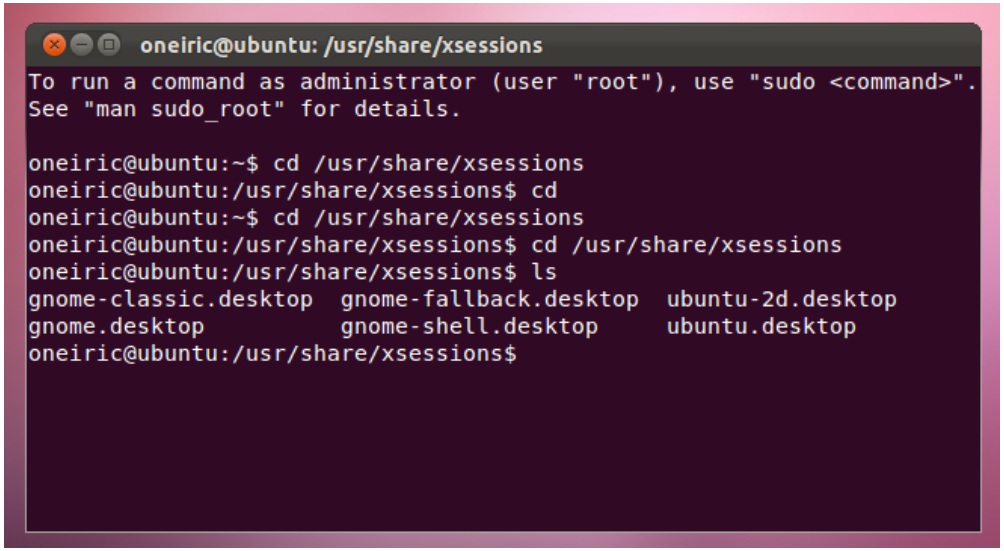
UNIX

- Three parts

Files and Processes	Everything in UNIX is a file or a process (programs are processes)
Shell	Interface between user and kernel, Command Line Interpreter (CLI)
Kernel	Central component, manages the computer's hardware

UNIX Shell

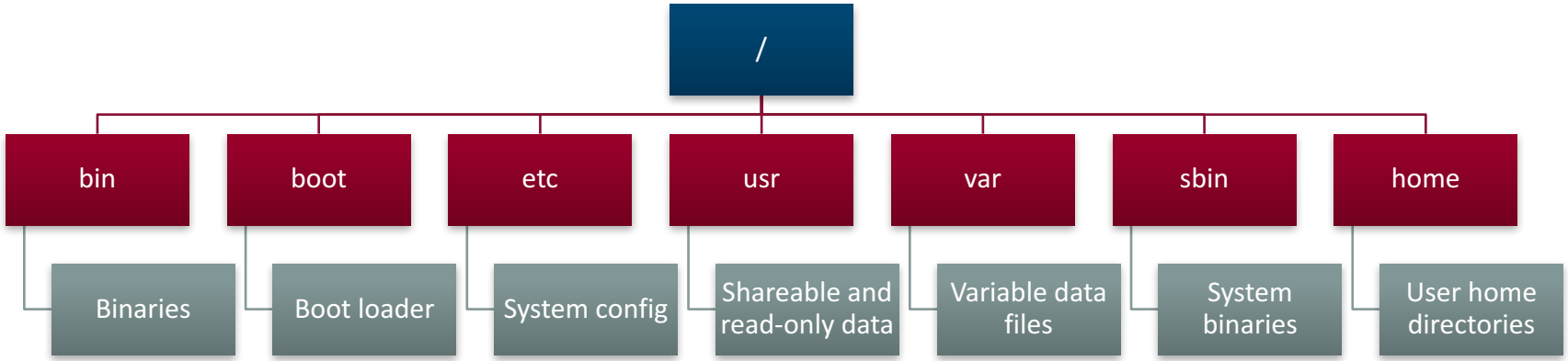
- Interprets commands
  - Commands are themselves programs
- Filename and command completion (use the [tab] key)
- History (use the cursor keys)
- Current working directory



Files and Processes

- Everything in UNIX is a file or a process
  - A process is an executing program (with a 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.tml`

Directory Structure



File Permissions

- Users
  - Unique username
  - Member of one or more groups
  - `/etc/passwd` and `/etc/group`
- Files
  - Owner (and associated group)
  - Set of permision 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` *`file`* – Owner can read/write, others can't do anything
  - `chmod +x` *`file`* – Add execute permission for everyone

Basic UNIX Commands - Browsing

`ls`	list files and directories
`ls -a`	list all files and directories
`ls f*`	list all files starting with 'f' (*: wildcard)
`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

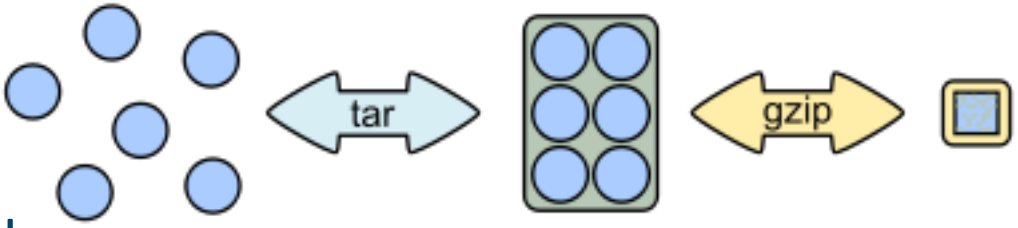
Basic UNIX Commands - Files

`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 (directory must be empty!)
`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 keyword
`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 Commands – Varia

`man` *`command`*	display the manual pages for a command
`data`	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`
  - List files: `tar -tf tarball.tar`
  - Upate files: `tar -uf tarball.tar 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
- kill current job: `Ctrl+C`

`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 users's jobs
`fg pid`	Bring process to the foreground
`bg pid`	Bring process to the background

Basic UNIX commands – 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
`ping` *`host`*	Send a small packet to a host (test reachability)

Useful UNIX commands – Screen

- When connected to a remote server, running processes will be killed when connection is lost or terminated.
- The command screen will start a virtual terminal, this screen is not linked to a particular terminal
- Screen session has the same behavior as a normal terminal

Outside a screen session	
`screen`	Start a new screen session
`screen -S` *`name`*	Start a new screen session with given name
`screen -r`	Resume a screen session
`screen -ls`	List all available screen sessions
`screen -X -S` *`sessionid`* `stuff` *`text`*	Send text as input to screen with given id
`screen -X -S` *`sessionid`* `quit`	Removes a screen session
Inside a screen session	
`Ctrl+a d`	Detach screen from the terminal

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

- Blackboard
- Course webpage
  - http://msdl.cs.mcgill.ca/people/hv/teaching/ComputerSystemsArchitecture/#CS1