Revision lecture on UNIX - 1

COMP1204: Data Management

About the COMP1204 exam

- Formative end-of-semester assessment
- Computer-aided tests on Blackboard, with multiple-choice questions

Example of type of multiple-choice questions

- You have to perform a task; identify the correct code snippet (from multiple options) to perform it.
- Identify the expected output for a given bash command.
- You would need to know what basic UNIX commands do (example: grep, chmod, wc, file etc.).
- You would also need to know basic regular expression syntax.
- All you would need are in the UNIX lecture slides.

Example question 1

You have compiled your C code to produce a binary file named *print-attendance.o*. Set the permissions for the following:

> You have full-access to the file.
>
> The group can only read and execute the file.
>
> Everyone else is allowed to only execute it.

1. chmod 640 *print-attendance.o*
2. chmod 751 *print-attendance.o*
3. None of the above

Example question 2

In your empty home directory you run the command

touch record-a.txt record-A.txt record-A.TXT RECORD-A.txt reCORD-A.txt

Now what will be the output of the following:

file * | wc –l

1. 1 as all the files have the same name
2. 5
3. None of the above

Note: The -l switch to wc prints the number of lines

Summary of our UNIX lectures

Summary of our UNIX lectures

1. Basic file navigation
2. Creating, manipulating and deleting files
3. Process management
4. File permissions
5. Wild cards and regular expressions
6. Grep, sed and awk

Basic file navigation

See slides at https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX2.pdf

- **pwd** print working directory, where we are currently located
- **ls** list content of directory
- **cd** change directory; move to another directory
- Remember these concepts:
  - ***Relative and Absolute paths***

Basic file navigation

See slides at https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX2.pdf

- **pwd** print working directory, where we are currently located
- **ls** list content of directory
- **cd** change directory; move to another directory
- Remember these concepts:
  - **Relative path**: Location of file or directory relative to our current location in file system.
  - **Absolute path**: Location of file or directory relative to root (**/**) of file system.

Creating, manipulating and deleting files

Commands for creation, manipulation and deletion

- **mkdir** Make Directory; Creates a new directory.
- **rmdir** Remove Directory; Deletes the directory.
- **touch** Create an empty file.
- **cp** Copy; Copies a file or directory.
- **mv** Move; Moves a file or directory (can also be used to rename file or directory).
- **rm** Remove; Deletes a file.

Commands for displaying file contents

See slides at https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX2.pdf

- Various commands are available if you want to just display a file in UNIX, i.e., *not edit it*:
  - **cat**
  - **less**
  - **head**
  - **tail**

Can anyone remember what ***head*** *does?*

Command: head and tail

- **head [options] <file name>** displays the top part of the file
- By default it shows the first 10 lines
- **-n** option allows you to change that
- Example: to displays the first 3 lines

[dst1m17@linuxproj ~]$ **head –n3 /home/dst1m17/COMP1204/airbnb/london/smallfile.csv**

- **tail [options] <file name>** is the same as **head**, but shows the last lines of the file.

[dst1m17@linuxproj ~]$ **tail –n3 /home/dst1m17/COMP1204/airbnb/london/smallfile.csv**

Process management

See slides at
https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX3.pdf

Commands for running a process in the background / foreground

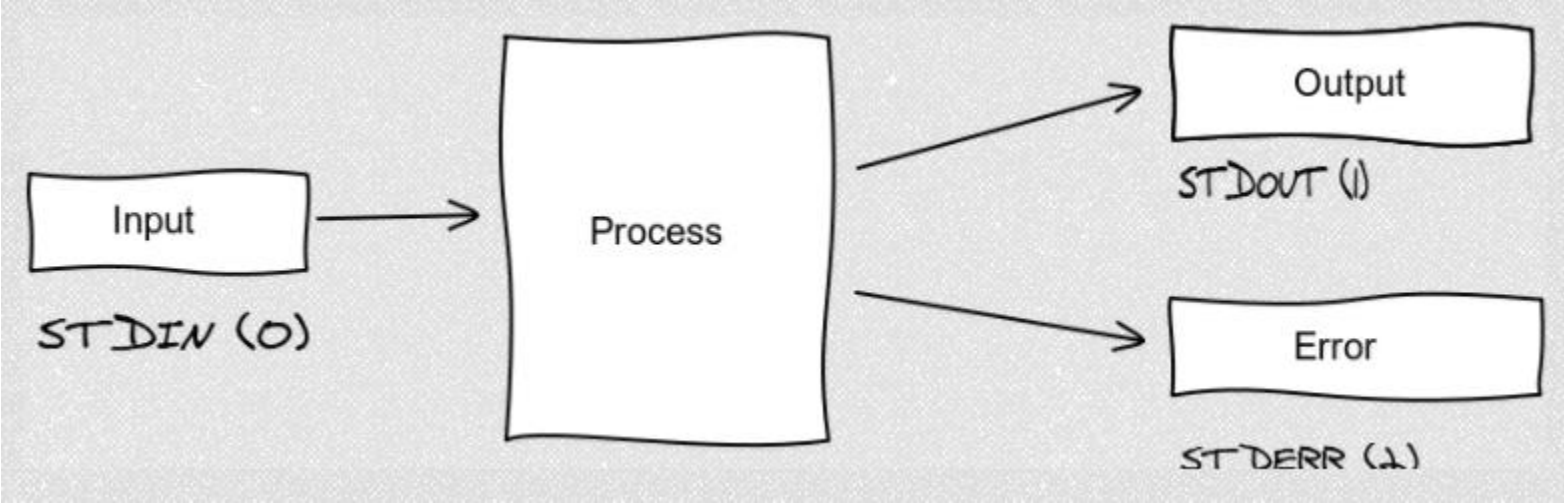
T

- **CTRL - z** Pause the current foreground process
- **bg** Move process to the background.
- **fg** Bring process back to the foreground.
- To continue running the process when logged off, you would need to use a window manager like **screen**

Pipes and filters

See slides at https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX3.pdf

Recap on basics of pipes Input/Output Redirection ("piping")



http://ynonp.github.io/unix2-bash-scripts-slides/#/18

- Bash shell allocates 3 file descriptors for each process
  - STDIN is opened for keyboard input
  - STDOUT, STDERR to screen output
- *We can change these!*

Input/Output Redirection (“piping”)

- Programs (or filters) can output to other programs
- Called “piping”
- **program_1 | program_2**
  - program_1’s output becomes program_2’s input
- **program_1 > file.txt**
  - program_1’s output and error logs are written to a file called “file.txt”
  - We can use **>>** instead of **>** to append to end of file.txt
- **program_1 < input.txt**
  - program_1 gets its input from a file called “input.txt”

Pipes and *filters*

- A filter is a program which accepts textual input and transforms it in some way.

- Filters can be connected together by pipes.

- Filters can be though of as building blocks to be easily put together to do what you want.

Filter 1 Filter 2 Filter 3

- Example: **cat** student_attendance.txt | **grep** –e August | **sort**

search for another month

sort in reverse order of attendance

Examples of piping with pipe operator |

The pipe operator (**|**) creates concurrently executing processes for each filter used.

[dst1m17@linuxproj london]$ **cat** tomslee*.csv **|** **cut -f4 -d,** **|** **sort** **|** **uniq** prints to screen

[dst1m17@linuxproj london]$ **cat** tomslee*.csv **|** **cut -f4 -d,** **|** **sort** **|** **uniq** **> processedlist.txt**

Wildcards to refer to multiple files

See slides at https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX4.pdf

Wildcard expansion

- Wildcards allow you to operate on multiple files at a time

- If the command-line argument has a wildcard, your shell – *the command line interpreter* -- will replace it with a list of matching filenames

Summary of wildcards for your revision

Wildcard	*Matches*
*	zero or more characters
?	exactly one character
[abcde]	exactly one character listed
[a-e]	exactly one character in the given range
[!abcde]	any character that is not listed
[!a-e]	any character that is not in the given range
{debian,linux}	exactly one entire word in the options given

http://ynonp.github.io/unix2-bash-scripts-slides/#/12

- Email me if you have any questions – dst1m17@soton.ac.uk

- Next UNIX revision session:
  - Recap on file permissions, bash scripts, grep, sed and awk.
  - Go over some example multiple-choice questions.