Basics on the Unix operating system

Inf553, 2018

Ioana Manolescu, Pierre Bourhis, Benoît Groz

Directories and quota

- The Unix file system can be viewed as a tree starting at **/**
- Each user has a **home directory**
- To know in which directory you are in a terminal, type **pwd**

("print working directory")

- To **copy** the content of a full directory /A/B under /A/C/D, type **cp –R /A/B /A/C/D/**

- Your home directories at Polytechnique have a **quota** (maximum size of all the files) of **2GB**
  - Thus, all your files for all your courses must fit in 2GB
- Avoid storing big files into your home directory
- Move files that are not used often in other places (e.g., flash drives, NFS, etc.)

Programs and processes

- A **program** is a file that someone has the rights to execute
  - Someone may or may not include you
  - To see who has the right to execute a file, type **ls –l file** and look for the "x":
    - Each file has read, write, execute rights for owner, owner's group, and others
    - r w x r - x - - x means: rwx for owner; rx for the owner's group; x for others
    - A program does not strictly speaking "run"; instead...
- To **run** a program, we launch a **process**, that is: one execution of an executable program
  - A process has a start and an end
  - There can be several executions of the same program at the same time, or successively

Programs and processes

- A **program** may exist as **several executable files**, in different locations
  - /usr/local/bin/eclipse and
  - /users/profs/2016/ioana.manolescu/SOFT/eclipse/eclipse
  - These **may or may not be the same version** of the program!
  - Important to be sure which one is running
  - Simple solutions:
    - launch it with the full path, or
    - go in the directory containing the executable program and launch it by typing ./program
- It pays to learn about environment variables and $PATH

Processes

- Any process has a **parent process** (typically the one that from which the process was started)
- When the Unix machine boots, the first process is automatically launched; it then has children processes, which have other children processes etc.
- When you log out, most of your processes end; the others continue running
  - System processes, remote sessions of other users etc.
- By default when you launch a process in a terminal, **the parent process does not interact with you until the child process is finished**
  - To keep interacting with the parent process, launch the child process in the **background**: **./program &**
  - The background child process may print things in the terminal but you can still use your terminal, send other commands etc.

Processes

- To **find out which processes are running which were started from your terminal**, type **ps**
- To find out more information about these processes, type **ps –l**
  - You will see the parent process ID (PPID), the process owner user ID (UID) etc.
- To find out extensive information about **all the processes on the machine**, type **ps –Al**
  - This will give lots of information!
  - To consume it at your own speed, see next slide

Processes

- Every process has an input, an output, and possibly some parameters
- Processes launched from the terminal will usually use the terminal as their output
- It is possible to **pipe (chain)** two processes so that the output of the first is given as input to the second: **program1 | program2**
- The **less** program, present on Unix, allows to see a file screen by screen (type whitespace to see the next page, q to quit)
- Therefore, to inspect a long program output, type: **program | less** for instance: **ps –Al | less**

Crucial tool: grep

- **grep** is an Unix tool which selects the lines of a file that contain a given string (regular expression)
- To find out all the processes running now the eclipse program, type:
  ps –Al | grep eclipse
  - Selects just the interesting lines from ps output

  Similarly **ps –Al | grep pg_ctl**
- It pays to learn about grep (man grep)

Crucial tool: find

- Finds all the files in a given directory that match a certain condition
- **find . –name *.java** finds all the files whose name ends in .java in the current directory or one of its subdirectories
- It pays to learn about find (man find)

Working with downloaded data

- When you are given **file.tgz**, good to know which kind of file this is
  - The file extension (.tgz) is just the way it was **named**
  - It *should* indicate what kind of file it is, but users may name things more or less correctly
- To learn what the file type really is: **file** command
    - Tells you the nature of the file and how to open it
    - Type: **file monfichier.tgz**
    - Then, **tar –xvf monfichier.tgz** usually extracts it in a directory

How to download a file without using a browser

- **wget –c URL**

For instance:

```
wget -c http://www-us.apache.org/dist/tomcat/tomcat-9/v9.0.1/bin/apache-tomcat-9.0.1.tar.gz
```