

## Introduction to Linux

### Objective

To be familiar and work with Linux using a standard terminal, which includes:

- Basic commands, text editors, and file system
- Passing arguments to commands
- Cascading multiple commands and passing data between them (piping)

Linux commands needed for the lab should all be included in the cheat sheet provided. The cheat sheet provides examples on how to use these commands. If you want to see all the options the command provides, consult the man page by typing in: ``man <name of the command>``

More commands can be found on the following website: <http://www.ee.surrey.ac.uk/Teaching/Unix/>

### Submission:

One compressed file that contains all the required output (below).

### Description

Most of the command line tools in Linux are written in C. A system administrator who masters the use of these commands can have a high degree of control of their OS and can manipulate data files with very little need for high-level programming skills.

These commands expect user inputs (arguments and flags) from the keyboard (as the standard input, **stdin**) and send their output to the screen (as the standard output, **stdout**). With piping, commands can be directed to accept input from other commands. Many system administration and data processing tasks can be achieved by piping a sequence of many commands.

The lab consists of the following parts (*see the hint at the end of the assignment*):

1. Data manipulation:

Please download the file “words” and write a file that contains all the commands needed to do the following:

- o Change the directory to where you just made the copy and print out the path to the directory.
  - o Print out the beginning of the file and the end of the file out.
  - o Search the file “words” for a word that was entered from the command line.
  - o Display how many lines are in the file “words”.
  - o Retrieve the beginning and the end of the file and then append the beginning to the end (think of using pipes).
  - o Retrieve the beginning, the end, and the word you searched earlier from the file then sort the results (think of using pipes).
-

## 2. Linux commands:

Write another file that contains all the commands needed to do the following:

- Run at least 10 more Linux commands (different from what you tested in part 1), capture their output in a format that shows the command and its results in an easy way (command-> new line-> output -> new line-> new line-> command, .... etc).
- At the top of your output file, make sure to use commands to print out the name of your user account and today's date. It should look something like this: "My Student Account is <your account> ", "Today's Date is <Date>".

**Hint:**

There are many ways to create a file to generate the required data. Below is one way to do this:

- create a file with all the commands;
- add statements that echo empty lines;
- change it mode to an executable program;
- run your batch program and direct its output to an output file using ">";
- The name of your account should be recovered by the USER variable. **Not hardcoded**
- Piping multiple commands into another command can be accomplished by:
  - { <1<sup>st</sup> command> ; <2<sup>nd</sup> command> ; ... ; <nth command> ; } | <receiving command>

## Other valuable resources:

- Use the site <https://www.howtoforge.com/> to search for Linux commands (see below)
    - <https://www.howtoforge.com/linux-pgrep-command/>
    - <https://www.howtoforge.com/tutorial/linux-find-command/>
    - <https://www.howtoforge.com/tutorial/linux-xargs-command/>
    - <https://www.howtoforge.com/linux-wc-command-explained-for-beginners-6-examples/>
  - Use the site <https://www.thegeekstuff.com/> for more resources (see below):
    - <https://www.thegeekstuff.com/2010/11/50-linux-commands/>
    - <https://www.thegeekstuff.com/2010/12/50-unix-linux-sysadmin-tutorials/>
    - <https://www.thegeekstuff.com/best-of-the-blog/>
-