

# Lab 1: SVN, Shell Use and Programming

## Introduction

The goal of the first lab is to provide you with a high level overview on how to use SVN, practice a few of the shell commands that you have seen in the lecture, learn a few new ones and learn about an extremely useful feature of shells, called pipes. To learn how the suggested shell commands work, run `man CMD` where `CMD` is the name of the command you want to learn about.

Remember, the TAs are there to help you. You should work in pairs and are welcome to ask other students for help too.

Here is a list of the shell commands that were mentioned in the lecture:

- `cd` -- change directory
- `pwd` -- print working directory
- `ls` -- list all files and sub-directories in a directory
- `stat` -- display information about a file
- `rm` -- remove (i.e. delete) a file
- `cp` -- copy a file
- `cat`, `more`, `less` -- list the contents of a file
- `chmod` -- change file mode, i.e. file permissions
- `ln` -- create a link
- `wc` -- count words

Here is a list of other shell commands that you might find useful:

- `mkdir` -- create a new directory
- `head` -- output only the first lines of a file
- `tail` -- output only the last lines of a file
- `grep` -- find a word in one or more files
- `ps` -- process status (lists running processes, often run as `ps aux` for the most information)
- `cut` -- extract a column from a file
- `sort` -- sort a file alphabetically
- `uniq` -- remove adjacent duplicate lines

## Getting ready to work

Create a directory (`mkdir`) in your home directory to do your lab work in. You might want to label it something identifiable like `cscb09labs`. Change into the directory (`cd`) you just created to do your work here.

```
~/cscb09_space
```

Copy the file `hockey_stats.txt` into your working directory:

```
cp /courses/courses/cscb09w19/nizamnu/labs/lab1/hockey_stats.txt .
```

(or download it from Quercus). The file contains a line for each of the 30 most successful players in the NHL (in terms of number of goals scored). The file has 23 columns, but we will use only the first 6 columns, which contain first name of the player, last name, team, position, games played, total goals. The file is sorted by number of goals scored.

## Warm-up exercises

These exercises provide some practice with the commands listed above and the use of man pages.

1. Return only the entries for the three top players (remember the file is already sorted by goals scored).
2. How many characters does the hockey file contain?
3. Sort the hockey players by the number of games played in decreasing order, i.e. starting with the players with the most games.
4. Filter out the entries for the players that played for Toronto.
5. Provide a list of the names of all players (nothing else) saved in a file called `players.txt`.

## Intro to Pipes

One extremely useful feature of shells are pipes. Pipes allow the pipeling of different shell commands by sending the output of one command to the input of another command. Pipes are created through the `|` symbol. For example, to count the number of lines produced by `ls -l` you can use the following pipeline:

```
ls -l | wc
```

You can also pipe more than two commands. For example, to display a sorted list of the last 10 lines of output of `ls -l` do:

```
ls | tail -n 10 | sort
```

## Exercises with pipes

1. Print out the names of all players sorted alphabetically by first name.
2. How many Toronto players are among the top 10 scorers?
3. Create a pipe that outputs only lines 10-20 of the hockey file.
4. What is the position of the player with the largest number of games played?
5. How many different teams have players that are listed in the hockey file?

6. Construct a pipe that counts the number of `sshd` processes running on a machine. You may want to `ssh` into `mathlab` to try this out. (In a shell type `ssh mathlab.utsc.utoronto.ca` and follow instructions.)
7. Next construct a pipeline that counts the number of users that are currently running an `sshd` process.

## Editors (Optional)

You will need to know at least one text editor to use on Unix systems. Only work through this section of the lab if you don't know how to use a text editor on Linux. If you already know how to use an editor, you can skip this.

One very simple editor is `nano`. You can open and edit the `hockey` file using `nano` by executing `nano hockey_stats.txt`. You can use the arrow keys to move your cursor in the file; `Ctrl-X` exits `nano` and returns you to the shell. Make a few modifications to your file, exit the editor and convince yourself using `cat` or more that your modifications have taken place.

Probably the most commonly used editor is `emacs`. Emacs is much more complex, but also much more powerful. If you are willing to invest some more time you might want to learn emacs.

There are many online tutorials. One example is this one:

<https://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html>

## Congrats on finishing the first lab of B09!