



FIT2100 Assignment #1
Building a Command Line Interpreter
with C Programming
Week 1 Semester 2 2017

Dr Jojo Wong
Lecturer, Faculty of IT.
Email: Jojo.Wong@monash.edu
© 2016-2017, Monash University

August 13, 2017

Revision Status

\$Id: FIT2100-Assignment-01.tex, Version 1.0 2017/08/13 21:30 Jojo \$

Contents

1	Introduction	4
2	Command Line Interpreter	5
2.1	Task 1: Basic Command Line Interpreter	5
2.2	Task 2: Advanced Command Line Interpreter	6
2.3	Important Notes	6
2.4	Sample Outputs	7
2.5	Marking Criteria	7
3	Submission	8
3.1	Deliverables	8
3.2	Academic Integrity: Plagiarism and Collusion	8

1 Introduction

This assignment is due on **1st September 2017 (Friday) by 5:00pm**. It is worth **10% of the total unit marks**. A penalty of 5% per day will be applied for late submission. Refer to the FIT2100 Unit Guide for the policy on extension or special consideration.

Note that this is **an individual assignment** and **must be your own work**. Please pay attention to Section 3.2 of this document on the University's policies on the *Academic Integrity, Plagiarism and Collusion*.

This first assignment consists of **two main tasks** and both tasks should be submitted as **two separate C programs** with supporting documentation on its usage. All the program files and any supporting documents should be compressed into one single file for submission. (The submission details are given in Section 3.)

Assessment: The first task and the second task carry an unequal weightage of marks with 60% and 40% respectively.

2 Command Line Interpreter

A *command line interpreter* (CLI) is a fundamental user interface to an operating system. Users direct the operations of a computer system by entering commands in the form of successive lines of text (i.e. command lines) for the command line interpreter to execute. In the Unix/Linux environment, the command line interpreter is often known as *the shell*.

The objective of this assignment is to implement a reduced version of a CLI program that has the similar functionality of the Unix/Linux shell. The program should be implemented on the Unix/Linux operating system using the C programming language with certain Unix commands.

2.1 Task 1: Basic Command Line Interpreter

In the first task, you are required to implement a basic CLI program which performs all of the following functions with the specific commands:

Command	Description
clear	Clear the output screen.
cd <directory>	Change the default directory to <directory>. If <directory> does not exist, an appropriate error message should be displayed. If the argument is not present, change the current directory to the directory given by the HOME environment variable in Unix. This command should also update the PWD variable.
ls <directory>	List the contents of the given directory named <directory>.
echo <statement>	Display <statement> on the output screen followed by a newline character '\n'. A statement is a sequence of characters begins with a double quote (") and terminated with the next double quote (").
help	Display the user manual using the command more or less in Unix.
quit	Terminate the CLI program and exit.

2.2 Task 2: Advanced Command Line Interpreter

In the second task, you should extend the basic CLI program implemented in Task 1 (Section 2.1) to support the following advanced functions based on the specific commands:

Command	Description
create <file>	Create a new file with the name <file>.
copy <old> <new>	Copy the contents of an existing file named <old> to a new file named <new>. If <old> does not exist, an appropriate error message should be displayed.
search <pattern> <file>	Display the number of occurrences of a sequence of characters <pattern> in the file named <file>. If <file> does not exist, an appropriate error message should be displayed.
run <program>	Execute an executable program named <program>.
halt <program>	Stop executing the program named <program>.

2.3 Important Notes

Your program should not display any prompt statements to ask for user commands. However, you may use a special character (symbol) to denote the command prompt as like the Unix shell prompt (e.g. \$). All the program outputs should be displayed on the standard output (i.e. the terminal screen).

Commenting your code is essential as part of the assessment criteria (refer to Section 2.5). You should also include comments at the beginning of your program file, which specify your name, your Student ID, the start date and the last modified date of the program, as well as with a high-level description of the program. In-line comments within the program are also part of the required documentation.

2.4 Sample Outputs

The following sample screen demonstrates a running CLI with a command prompt represented by the character '\$'. It is waiting for user input.

```
1 $
```

When the command `echo "FIT2100 first assignment"` is entered, the output screen should display "FIT2100 first assignment" as a new line.

```
1 $ echo "FIT2100 first assignment"
2 $ FIT2100 first assignment
```

Similarly, when the command `clear` is entered, all the outputs on the screen should be cleared.

```
1 $ echo "FIT2100 first assignment"
2 $ FIT2100 first assignment
3 $ clear
```

```
1 $
```

2.5 Marking Criteria

The assessment of this assignment will be based on the following marking criteria. The same marking criteria will be applied on both tasks (Task 1 and Task 2):

- 60% for working program;
- 10% for code architecture (algorithms, use of procedures and libraries, etc.);
- 10% for coding style (clarity in variable names, function names, etc.);
- 20% for documentation (both program comments and user documentation).

3 Submission

There will be NO hard copy submission required for this assignment. You are required to submit your assignment as a .zip file named with your Student ID. For example, if your Student ID is 12345678, you would submit a zipped file named 12345678_A1.zip. Note that marks will be deducted if this requirement is not strictly complied with.

Your submission is via the assignment submission link on the FIT2100 Moodle site by the deadline specified in Section 1, i.e. **1st September 2017 (Friday) by 5:00pm**.

3.1 Deliverables

Your submission should contain the following documents:

- A completed the assignment submission statement for online submission via the FIT2100 Moodle site.
- An user documentation (not more than 3 pages) in PDF format with clear and complete instructions on how to run your programs. (Note that your programs must at least run on the Unix/Linux system in the University's computer labs. Any submission that does not run accordingly will receive no marks.)
- Electronic copies of ALL your files that are needed to run your programs.

Marks will deducted for any of these requirements that are not strictly complied with.

3.2 Academic Integrity: Plagiarism and Collusion

Plagiarism Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass them off as your own by failing to give appropriate acknowledgement. This includes materials sourced from the Internet, staff, other students, and from published and unpublished works.

Collusion Collusion means unauthorised collaboration on assessable work (written, oral, or practical) with other people. This occurs when you present group work as your own or as the work of another person. Collusion may be with another Monash student or with people or students external to the University. This applies to work assessed by Monash or another university.

It is your responsibility to make yourself familiar with the University's policies and procedures in the event of suspected breaches of academic integrity. (Note: Students will be asked to attend an interview should such a situation is detected.)

The University's policies are available at: <http://www.monash.edu/students/academic/policies/academic-integrity>