

CSCI 3120 Operating Systems

Assignment 1: A Simple C Program for Linux

Due: 16:00, Sept. 22, 2023

- **Teaching Assistants:**
 - Kamran Awaisi (km521977@dal.ca)
 - Hui Huang (huihuang@dal.ca)
 - Arka Ghosh (arka.ghosh@dal.ca)
 - Yitong Zhou (yt760204@dal.ca)
 - Shiyun Wang (sh776410@dal.ca)
- **Help Hours: FCS Learning Center (Goldberg 233) or via the Channel “Office Hour - TAs” on MS Teams:**
 - Monday: 1:00pm-2:00pm, Kamran Awaisi
 - Tuesday: 1:00pm-2:00pm, Hui Huang
 - Wednesday: 1:00pm-2:00pm, Arka Ghosh
 - Thursday: 1:00pm-2:00pm, Yitong Zhou
 - Friday: 1:00pm-2:00pm, Shiyun Wang

1. Assignment Overview

In this assignment, you need to design and implement a C program that prompts the user to enter a sentence, splits the sentence into multiple words, then displays the words on the screen and places the words in a new file. Your program will be tested on the undergraduate server “csci3120.cs.dal.ca”, which is a Linux machine.

2. Important Information

There is a [zero-tolerance policy on academic offenses](#) such as plagiarism or inappropriate collaboration. By submitting your solution for this assignment, you acknowledge that the code submitted is your own work. You also agree that your code may be submitted to a plagiarism detection software (such as MOSS) that may have servers located outside Canada [unless you have notified me otherwise, in writing, before the submission deadline](#). Any suspected act of plagiarism will be reported to the Faculty’s Academic Integrity Officer in accordance with Dalhousie University’s regulations regarding Academic Integrity. Please note that:

1) The assignments are individual assignments. You can discuss the problems with your friends/classmates, but you need to write your program by yourself. There should not be much similarity between your program and others’ programs.

- 2) When you refer to some online resources to complete your program, you need to understand the mechanism, then write your own code. Your code should not be similar to the online resources. In addition, you should cite the sources via comments in your program.
- 3) You may use AI-driven tools to assist your learning, but you are not allowed to use them to produce work to be submitted for course evaluations. Your submitted program should not be highly similar to others' programs.

3. Detailed Requirements

1) Overview: In this assignment, you need to design and implement a C program that prompts the user to enter a sentence, splits the sentence into multiple words, then displays the words on the screen and places the words in a new file. The following steps should be implemented in your program:

Step 1: Prompt Displaying: When your program is executed, the following prompt should be displayed on the screen, asking the user to enter a sentence. Then the user could enter a sentence on the same line.

Please Enter Your Sentence:

Please note that:

- The user sentence is composed of multiple words (i.e. 2 or more words).
- There is only one space between neighboring words.
- In this assignment, each word is composed of a series of symbols (i.e. 2 or more symbols) and each symbol in a word can be a letter or a numeric digit (i.e. 0, 1, ..., 9) or a punctuation mark that can be found on a keyboard.
- You can assume that the user input is composed of 1 symbol at the least and 200 symbols at the most. This range for user input (i.e. 1-200 symbols) applies to spaces. Namely, each space is counted as a symbol in the user input.
- In this assignment, you can assume that the user input always satisfies the requirements mentioned above. Namely, your program does not need to do format check

Step 2: Input Processing: After the user enters a sentence and presses the return key, your program receives the input and splits the user input into multiple words. Then the words are displayed on the screen.

Please note that:

- There should be only one word in each line and there should be no empty lines.
- In addition, the order of the words in the user input should be preserved. For example, if the user enters "We all like computer science", then the word "We" should appear in the first line and the word "all" should appear in the second line, etc.
- After your program is executed, the command prompt and the last word of the user input should not be on the same line. Namely, the command prompt should be on a line next to the line including the last word of the user input.

Step 3: File Creation: Thereafter, your program needs to create a new file named “Output.txt” in the directory where your program is located. You can assume that no file named “Output.txt” exists in the same directory as your program.

Step 4: Output Processing: Once the file “Output.txt” is created, your program will put the words that were obtained previously in the file. Note that there is only one word in each line and there should be no empty lines. In addition, the order of the words in the user input should be preserved. For example, if the user enters “We all like computer science”, then the word “We” should appear in the first line of “Output.txt” and the word “all” should appear in the second line, etc.

Please note that:

- a) In Step 2, you need to display the words on the screen. In Step 4, you need to write the same set of words into the output file. These two steps could be carried out using the same loop in your program. Namely, after finding out a word in the user input, your program can display the word on the screen and thereafter write the word into the output file. Of course, if you prefer to use two separate loops to implement these two steps, it is also acceptable.
- b) In the file “Output.txt”, there should be a newline character at the end of the line that includes the last user word. With this newline character, when you run the command “cat Output.txt” on “csci3120.cs.dal.ca”, the last user word and the command prompt will not be on the same line (i.e. the command prompt will be on the line next to the line including the last user word).
- c) “csci3120.cs.dal.ca” is the computer used by the TA to evaluate your program. Therefore, you need to make sure that your program works on “csci3120.cs.dal.ca”.
 - a. You could use your CS ID to log on to “csci3120.cs.dal.ca” in order to write your program. Alternatively, you can write your program on other machines (e.g. your own computer), then transfer your program to “csci3120.cs.dal.ca” and thereafter test it on “csci3120.cs.dal.ca”.
 - b. If you do not know your CS ID, you can visit the following webpage to get your CS ID. If your CS ID does not work or you have a question about your CS ID, please send an email to “cshelp@cs.dal.ca”.
<https://csid.cs.dal.ca/>
- d) Essentially, the user input is a series of symbols with the new line character (i.e. ‘\n’) at the end. There are a variety of different functions in the GNU C library (i.e. glibc) that you can use to get the user input. One simple method is based on fgets(). Here are two links about fgets():
 - a. <https://www.geeksforgeeks.org/fgets-gets-c-language/>
 - b. <https://man7.org/linux/man-pages/man3/fgets.3p.html>Another possible method is based on scanf(), here is a link about scanf():
https://www.tutorialspoint.com/cprogramming/c_input_output.htm
- e) User input splitting: There are a variety of different ways to parse user input and generate the required words. One simple way is to utilize “strtok()”. Here are two links about this function:
 - a. <http://www.cplusplus.com/reference/cstring/strtok/>

- b. <http://man7.org/linux/man-pages/man3/strtok.3.html>

Note that:

- a. strtok() breaks a string into a sequence of zero or more nonempty tokens. On the first call to strtok(), the string to be parsed should be specified as a parameter. In each subsequent call that parses the same string, the corresponding parameter must be NULL.
 - b. strtok() could adopt multiple delimiters. For example, “,-” is a string including four delimiters (note that the first delimiter is a space character); “\n” is a string including two delimiters (note that the second delimiter is a new line character). One example illustrating how to use multiple delimiters can be found here: <http://www.cplusplus.com/reference/cstring/strtok/>
- f) Here are two useful webpages, which discuss C strings and how to use an array of pointers to character to store a list of strings:
- a. https://www.tutorialspoint.com/cprogramming/c_strings.htm
 - b. https://www.tutorialspoint.com/cprogramming/c_array_of_pointers.htm
- g) Your program must use the GNU C library (i.e. glibc) to implement the required features. Here are a few links about glibc:
- a. <https://www.gnu.org/software/libc/documentation.html>
 - b. <https://www.kernel.org/doc/man-pages/>
 - c. <http://man7.org/linux/man-pages/index.html>
- h) Compiling and running your program on “csci3120.cs.dal.ca” should not lead to errors or warnings. To compile and run your program on “csci3120.cs.dal.ca”, you need to be able to access the command-line interface of “csci3120.cs.dal.ca”. In addition, you need to be able to upload a file to or download a file from “csci3120.cs.dal.ca”.
- a. To access the command-line interface of “csci3120.cs.dal.ca”, several different methods could be used. Here are two widely used methods:
 - i. MS Windows Computer: You can use the software tool “putty” on MS Windows computers. With “putty”, the connection type should be set to “SSH”. Note that “putty” can be downloaded via the following link: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>.
 - ii. Mac and Linux Computer: On Mac and Linux computers, you can use the command “ssh” to access “csci3120.cs.dal.ca” via the program called “Terminal”.
 - b. To transfer files between your computer and “csci3120.cs.dal.ca”, several different methods could be used. Here are two widely used methods:
 - i. MS Windows and Mac Computer: Cyberduck is popular tool used to transfer files between two computers via SFTP (i.e. SSH File Transfer Protocol). You can download Cyberduck from the following webpage: <https://cyberduck.io/>. This document includes an appendix that provides an introduction to Cyberduck.
 - ii. Linux Computer: On Linux computers, you can use the command “scp” to transfer files. Here is a tutorial on the command “scp”: <https://www.linuxtechi.com/scp-command-examples-in-linux/>.
 - c. You should upload your program to “csci3120.cs.dal.ca” to verify that it works well on “csci3120.cs.dal.ca”. After verifying that your program works on

"csci3120.cs.dal.ca". You should submit your assignment in brightspace. For submission details, please proceed to the submission section of this document. Note that the TA will NOT look at your files on "csci3120.cs.dal.ca". Instead, the TA will retrieve your submission from brightspace.

2) Readme File: You need to complete a readme file named "Readme.txt", which includes the instructions that the TA could use to compile and execute your program on "csci3120.cs.dal.ca".

3) Submission: Please pay attention to the following submission requirements:

- a) You should place "Readme.txt" in the directory where your program files are located.
- b) Your program files and "Readme.txt" should be compressed into a zip file named "YourFirstName-YourLastName-ASN1.zip". For example, my zip file should be called "Qiang-Ye-ASN1.zip".
- c) Finally, you need to submit your zip file for this assignment via the submission link in brightspace. As mentioned previously, the TA will NOT look at your files on "csci3120.cs.dal.ca". Instead, the TA will retrieve your submission from brightspace.

Note that there is an appendix at the end of this document, which includes the commands that you can use to compress your files on "csci3120.cs.dal.ca". In addition, there is another appendix, which illustrates how to view and kill your processes on "csci3120.cs.dal.ca".

4. Grading Criteria

The TA will use your submitted zip file to evaluate your assignment. The full grade is 10 points. Your submission will be evaluated from the following perspectives. The specific rubric used by the marker is included in a separate file.

- a) "Readme.txt" with the correct compilation/execution instructions is provided. [1 Point]
- b) *Prompt Displaying*: When your program is executed, the following prompt should be displayed on the screen, asking the user to enter a sentence. Then the user could enter a sentence on the same line. [1 Points]
- c) *Input Processing*: After the user enters a sentence and presses the return key, your program receives the input and splits the user input into multiple words. Then the words are displayed on the screen. [4 Points]
- d) *File Creation*: Thereafter, your program needs to create a new file named "Output.txt". [1 Points]
- e) *Output Processing*: Once the file "Output.txt" is created, your program will put the words that were obtained previously in the file. Note that there is only one word in each line. In addition, the order of the words in the user input should be preserved. [2 Points]
- f) Proper programming format/style (e.g. proper indentation, proper variable names, proper comments, release the memory that is dynamically allocated when it is not needed any more, etc.). [1 Point]

Please note that when “Readme.txt” is not provided or “Readme.txt” does not include the compilation/execution instructions, your submission will be compiled using the standard command `gcc -o A1 A1.c` (or your filename), and [you will receive a zero grade if your program cannot be successfully compiled on “csci3120.cs.dal.ca”](#).

5. Academic Integrity

At Dalhousie University, we respect the values of academic integrity: honesty, trust, fairness, responsibility and respect. As a student, adherence to the values of academic integrity and related policies is a requirement of being part of the academic community at Dalhousie University.

1) What does academic integrity mean?

Academic integrity means being honest in the fulfillment of your academic responsibilities thus establishing mutual trust. Fairness is essential to the interactions of the academic community and is achieved through respect for the opinions and ideas of others. Violations of intellectual honesty are offensive to the entire academic community, not just to the individual faculty member and students in whose class an offence occur (See Intellectual Honesty section of University Calendar).

2) How can you achieve academic integrity?

- Make sure you understand Dalhousie’s policies on academic integrity.
- Give appropriate credit to the sources used in your assignment such as written or oral work, computer codes/programs, artistic or architectural works, scientific projects, performances, web page designs, graphical representations, diagrams, videos, and images. Use RefWorks to keep track of your research and edit and format bibliographies in the citation style required by the instructor. (See <http://www.library.dal.ca/How/RefWorks>)
- Do not download the work of another from the Internet and submit it as your own.
- Do not submit work that has been completed through collaboration or previously submitted for another assignment without permission from your instructor.
- Do not write an examination or test for someone else.
- Do not falsify data or lab results.

These examples should be considered only as a guide and not an exhaustive list.

3) What will happen if an allegation of an academic offence is made against you?

I am required to report a suspected offence. The full process is outlined in the Discipline flow chart, which can be found at:

<http://academicintegrity.dal.ca/Files/AcademicDisciplineProcess.pdf> and includes the following:

- a. Each Faculty has an Academic Integrity Officer (AIO) who receives allegations from instructors.

- b. The AIO decides whether to proceed with the allegation and you will be notified of the process.
- c. If the case proceeds, you will receive an INC (incomplete) grade until the matter is resolved.
- d. If you are found guilty of an academic offence, a penalty will be assigned ranging from a warning to a suspension or expulsion from the University and can include a notation on your transcript, failure of the assignment or failure of the course. All penalties are academic in nature.

4) Where can you turn for help?

- If you are ever unsure about ANYTHING, contact myself.
- The Academic Integrity website (<http://academicintegrity.dal.ca>) has links to policies, definitions, online tutorials, tips on citing and paraphrasing.
- The Writing Center provides assistance with proofreading, writing styles, citations.
- Dalhousie Libraries have workshops, online tutorials, citation guides, Assignment Calculator, RefWorks, etc.
- The Dalhousie Student Advocacy Service assists students with academic appeals and student discipline procedures.
- The Senate Office provides links to a list of Academic Integrity Officers, discipline flow chart, and Senate Discipline Committee.

Appendix 1: How to Use Zip and Unzip on “csci3120.cs.dal.ca”

To compress:

```
zip squash.zip file1 file2 file3
```

To uncompress:

```
unzip squash.zip
```

this unzips it in your current working directory.

Appendix 2: How to View/Kill Your Processes on “csci3120.cs.dal.ca”

Due to programming mistakes, you might leave a number of running processes on “csci3120.cs.dal.ca”. When you leave too many running processes “csci3120.cs.dal.ca”, your CS ID could be locked temporarily (then you need to talk to CS Help to unlock your account). To view/kill your processes on “csci3120.cs.dal.ca”, you can use the following commands. Please keep an eye on your processes on “csci3120.cs.dal.ca”.

- 1) Command to display the processes belonging to a specific ID (i.e. UID): **ps -u UID**
- 2) Command to kill a process using process ID (i.e. PID): **kill -9 PID**
- 3) Command to kill all processes belonging to a specific user ID (i.e. UID): **pkill -u UID**

Appendix 3: Introduction to Cyberduck

1. Optional Registration: Cyberduck is free software that supports file transfers via SSH File Transfer Protocol (i.e. SFTP). You can choose to pay for the registration to avoid seeing the donate/buy window when you quit the program. But it is perfectly ok to use Cyberduck without the registration. The only drawback is that you will see the donate/buy window when it is terminated (in this case, you can click on “Later” to completely terminate the program).
2. Set Up Connection: SFTP (SSH File Transfer Protocol) is used to set up the connection
 - a. Start Cyberduck
 - b. Click on the icon “Open Connection”
 - c. Set Protocol to “SFTP (SSH File Transfer Protocol)”
 - d. Set Server to “csci3120.cs.dal.ca”
 - e. Enter your CSID and password
 - f. Check “Add to keychain” on macOS or check “Save Password” on MS Windows
 - g. Click on “Connect”
3. Upload/Download Files: “drag-and-drop” is supported
 - a. Uploading: select a file on your computer and thereafter drag the file into the Cyberduck interface
 - b. Downloading: select a file in the Cyberduck interface and thereafter drag the file to the proper destination