

CS314-001 Operating Systems, Spring 2024

Project #0 Description

Due: **9:30 a.m.**, January **18**, 2024

I. Project Narrative: The primary objectives in this project are:

- (1) confirm your account (user name and password) in the OS server (a UNIX-based time-sharing server we use to complete the programming projects in this course)
- (2) login to your account in the OS server using a SSH telnet tool (e.g., “Putty”)
- (3) learn how to use a text editor and write/edit a C source code file using a text editor (e.g., “*vi*” and “*emacs*”). For uploading existing files to your UNIX account (instead of writing C source code files by yourself), use “WinSCP”
- (4) compile your C source code file to a binary executable, using a UNIX standard C compiler, “cc”
- (5) execute your binary executable
- (6) learn basic UNIX text-based commands

II. Required Activities:

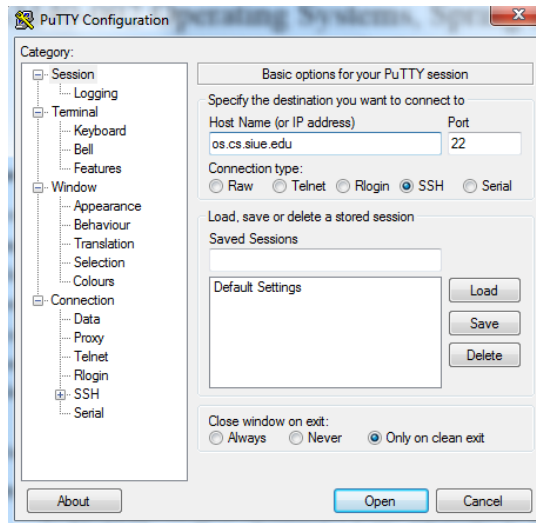
There are two parts in this project. For each part, the following activities are required for full credit:

PART I: compiling a C source code file using “cc” UNIX standard C compiler (**15 minutes**)

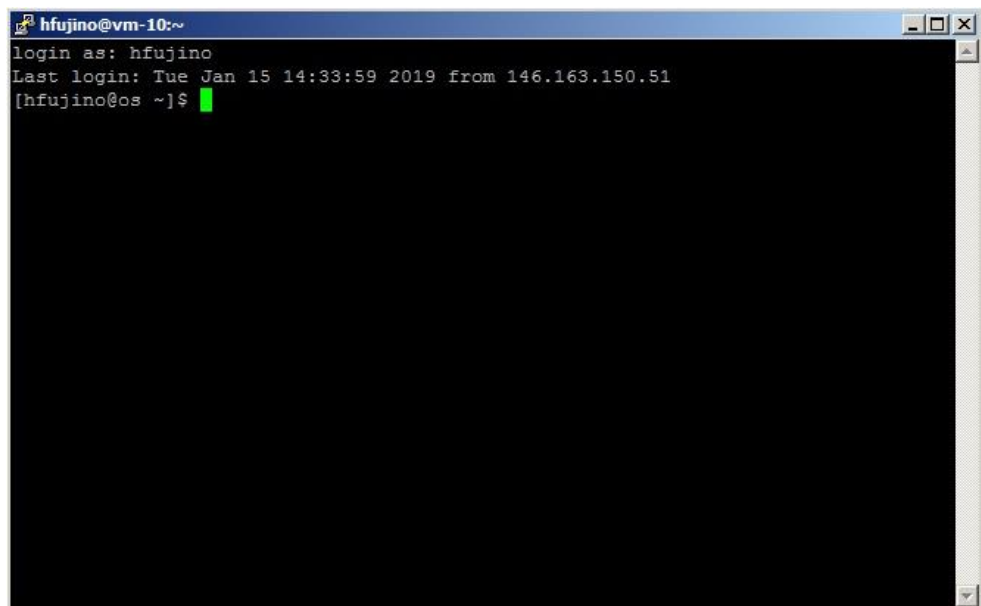
Step 1: Download “Putty” (a SSH telnet client) to your PC. First manually create a directory for “Putty” (anywhere in your PC) and download “putty_beta_0_66.exe” from the suggested web site (its URL is attached at the end of this document) or from the course home page to the directly. After you download the Putty.exe file to the directory, just double-click its program icon to start “Putty”.

Step 2: Login to your account using Putty: when you start “Putty”, use the following parameter to establish a SSH telnet connection to the OS server:

- **Host name:** “os.cs.siu.edu”
- **Connection type:** “SSH”
- **Port:** 22 (port 22 will be automatically set up as soon as you select “SSH” for connection type.



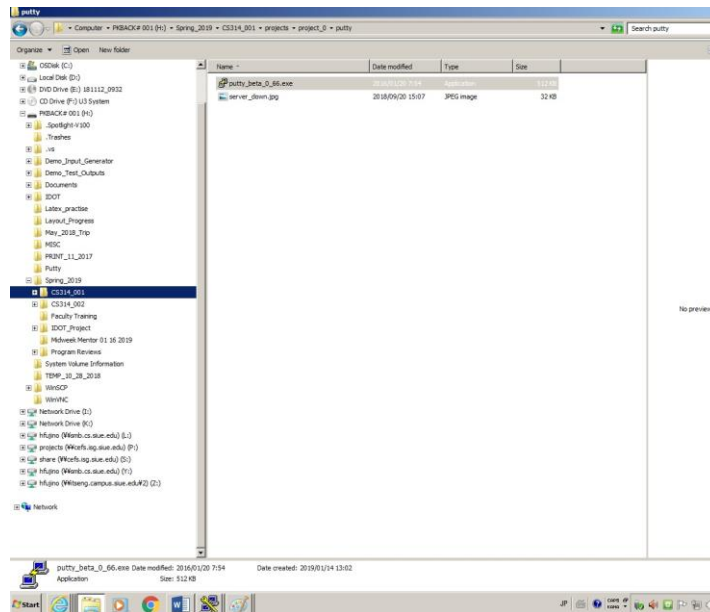
Then, press “Open”. After your press “Open”, an authentication screen will show up as shown below ([Figure 1](#)).



For your user name, enter your SIUE username. For your password, enter your SIUE password. On a successful login, a text-based command prompt will show up (as shown above).

Step 3: Prepare your C source code file. There are a couple different methods to perform this activity:

Method 1: Create your C source code file using a text editor (e.g., “notepad”) in your Windows and upload it to your OS server account. You can upload your C source code file using Windows Explorer (your OS server account is mapped to (“\\vm-00” network drive) or a file transfer tool, such as WinSCP (use whichever way easier for you). In this project, this method is suggested to save your time.



Method 2: Type the source code file by using a built-in standard text editor (e.g., “emacs” and “vi”). The following is an example for “vi” text editor.

```
hfujino@vm-10:~/CS314_519/project_00
# *****
# project0.c:
#
# compile: "cc project0.c"
# host: os.cs.siu.edu
# Hiroshi Fujinoki (hfujino@siue.edu)
# *****
#include <stdio.h>

int main()
{
    char my_name[20] = "Hiroshi Fujinoki";

    int SIUE_ID = 123456789;

    printf("\n");
    printf("My name is %s \n", my_name);
    printf("SIUE ID is %d\n", SIUE_ID);
    printf("\n");

    return 0;
}
```

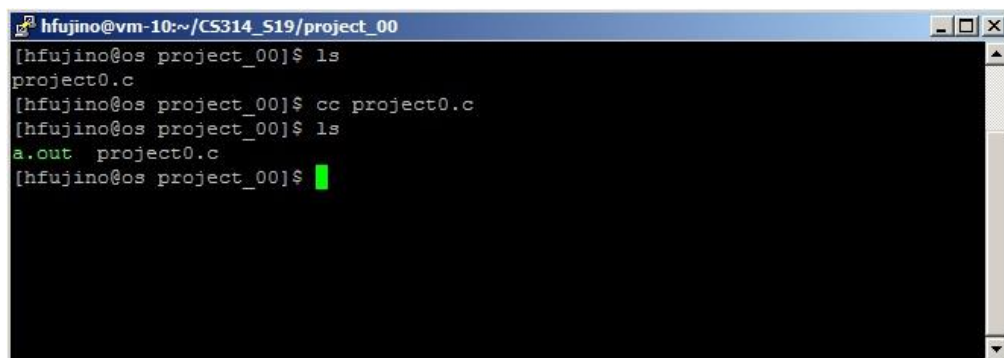
Step 4: Compile your C source code file using “cc” UNIX standard C compiler, using the following command.

```
cc project0.c
```

If you receive a compile-time error, fix the error and compile it again.

Step 5: Execute your binary executable.

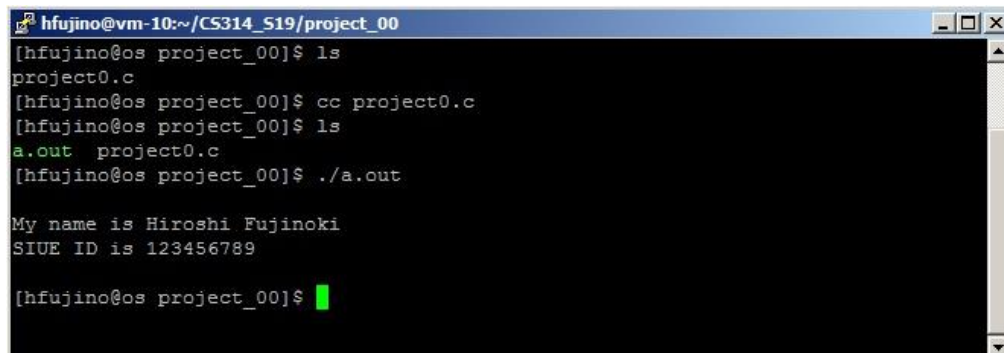
After you successfully compile your C source code file, the “cc” compiler should generate a binary executable (“a.out”). You can double-check it using “ls” command ([Figure 2](#)).

A terminal window titled 'hfujino@vm-10:~/CS314_S19/project_00'. The terminal shows the following commands and output:

```
[hfujino@os project_00]$ ls
project0.c
[hfujino@os project_00]$ cc project0.c
[hfujino@os project_00]$ ls
a.out project0.c
[hfujino@os project_00]$
```

To execute your binary executable ([Figure 3](#)):

```
“./a.out”
```

A terminal window titled 'hfujino@vm-10:~/CS314_S19/project_00'. The terminal shows the following commands and output:

```
[hfujino@os project_00]$ ls
project0.c
[hfujino@os project_00]$ cc project0.c
[hfujino@os project_00]$ ls
a.out project0.c
[hfujino@os project_00]$ ./a.out

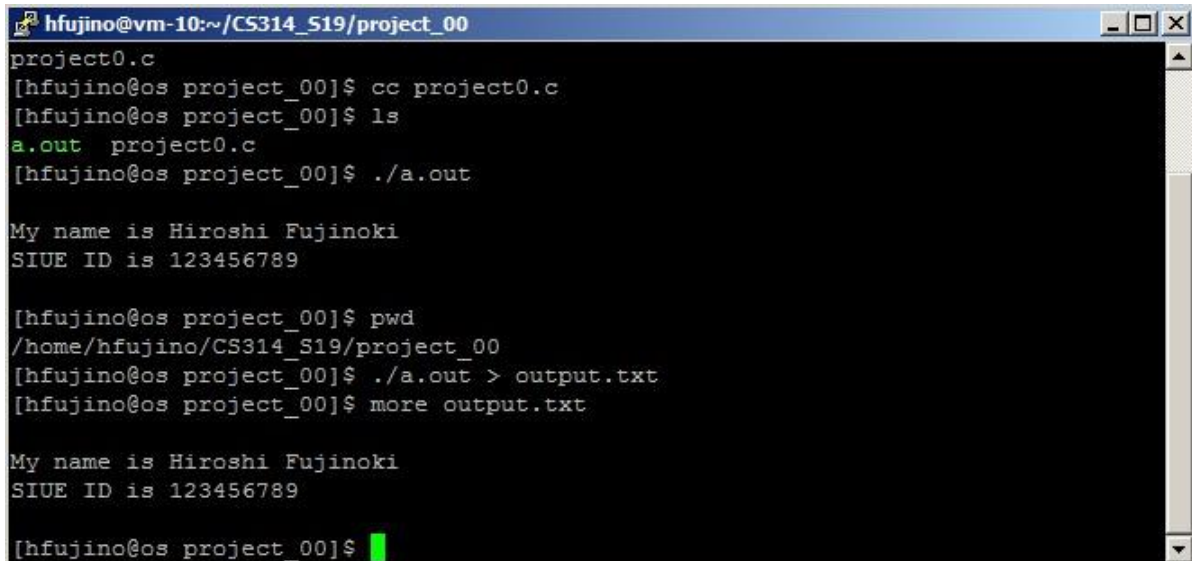
My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$
```

PART II: Using other C commands, such as *pwd*, *more*, *grep*, *mv*, and *diff* (15 minutes)

After “PART I”, do the followings (in that order):

Step 1: First, execute “*pwd*” to make sure where (in which directory) you are currently in. Then, execute “*./a.out > output.txt*” (this operation is called “output redirection”). After that, execute “*more output.txt*”. Capture your screen snapshot then (and save it as “[Figure 4](#)”).



```
hfujino@vm-10:~/CS314_S19/project_00
project0.c
[hfujino@os project_00]$ cc project0.c
[hfujino@os project_00]$ ls
a.out project0.c
[hfujino@os project_00]$ ./a.out

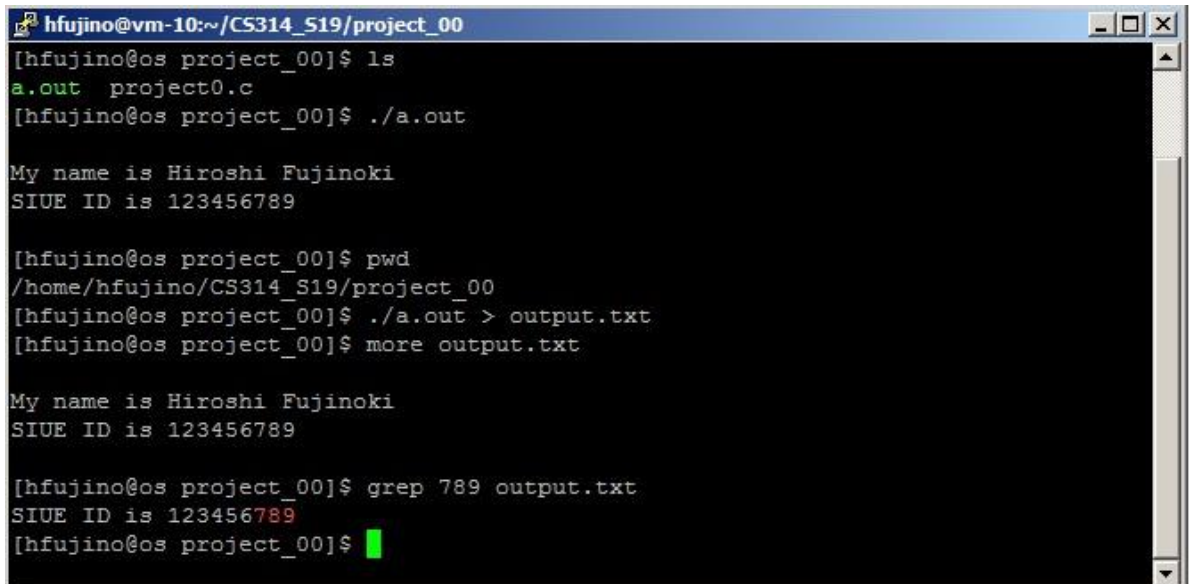
My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$ pwd
/home/hfujino/CS314_S19/project_00
[hfujino@os project_00]$ ./a.out > output.txt
[hfujino@os project_00]$ more output.txt

My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$
```

Step 2: Execute “*grep nnn*” (“*nnn*” is your “last three digits”). Then, execute “*mv output.txt output_01.txt*”. Then, “*ls*”. Capture your screen snapshot then (and save it as “[Figure 5](#)”).



```
hfujino@vm-10:~/CS314_S19/project_00
[hfujino@os project_00]$ ls
a.out project0.c
[hfujino@os project_00]$ ./a.out

My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$ pwd
/home/hfujino/CS314_S19/project_00
[hfujino@os project_00]$ ./a.out > output.txt
[hfujino@os project_00]$ more output.txt

My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$ grep 789 output.txt
SIUE ID is 123456789
[hfujino@os project_00]$
```

Step 3: Execute your “a.out” one more time with output redirection (as “./a.out > output_02.txt”). Then, execute “*diff* output_01.txt output_02.txt” and “*diff* output1.txt project1.c”. Capture your screen snapshot then (and save it as “[Figure 6](#)”).



```
hfujino@vm-10:~/CS314_S19/project_00
a.out project0.c
[hfujino@os project_00]$ ./a.out
My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$ pwd
/home/hfujino/CS314_S19/project_00
[hfujino@os project_00]$ ./a.out > output.txt
[hfujino@os project_00]$ more output.txt
My name is Hiroshi Fujinoki
SIUE ID is 123456789

[hfujino@os project_00]$ grep 123 output.txt
SIUE ID is 123456789
[hfujino@os project_00]$ mv output.txt output1.txt
[hfujino@os project_00]$ ls
a.out  output1.txt  project0.c
[hfujino@os project_00]$ ./a.out > output2.txt
[hfujino@os project_00]$ diff output1.txt output2.txt
[hfujino@os project_00]$ diff output1.txt project0.c
0a1,9
> /* *****
> * project0.c:
> *
> * compile: "cc project0.c"
> * host: os.cs.siue.edu
> * Hiroshi Fujinoki (hfujino@siue.edu)
> *
> * ***** */
> #include <stdio.h>
2,3c11,20
< My name is Hiroshi Fujinoki
< SIUE ID is 123456789
---
> int main()
> {
>     char my_name[20] = "Hiroshi Fujinoki";
>
>     int SIUE_ID = 123456789;
>
>     printf("\n");
>     printf("My name is %s \n" , my_name);
>     printf("SIUE ID is %d\n" , SIUE_ID);
>     printf("\n");
4a22,23
>     return 0;
> }
[hfujino@os project_00]$
```

III. Submission:

- (a) Your C source code file (your name and e-mail address should appear in the program header at the beginning of your C source code file). Your source code file should be saved as “project0_your3digits.c”.
- (b) The snapshots of your progress (Figure 1, 2, 3, 4, 5, and 6) saved as “Fig1.jpg”, “Fig2.jpg”, and so on.
- (c) All the seven files (the C source code file and the six jpeg image files) should be bundled in a zip file, using “project0_your3digits.zip” as the file name.
- (d) E-mail your *.zip archive file to the TA (the e-mail address of the TA will be announced). Specify “Section 001” in the subject field of your e-mail.
- (e) The submission (by e-mail) due is at 9:30 a.m. on January 18, 2024.
- (f) No late submission will be accepted for this project.

IV. Suggested References:

- UNIX commands: <https://www.tjhsst.edu/~dhyatt/superap/unixcmd.html>
- Text editors for UNIX: <https://www.unixtutorial.org/reference/unix-text-editors>
- WinSCP: <https://en.wikipedia.org/wiki/WinSCP>
- Putty SSH telnet client: <https://www.putty.org/>