

Exercise 1 – Implementing a Stack

Due by the end of your Week-2 laboratory class.

(2 marks)

This exercise is to be completed and submitted by the end of your week 2 laboratory class. After you complete and submit your work you must show your work and the submit receipt to your lab tutor to receive your marks. The submit instructions are given below and on the following page. The marking is based mainly on correct implementation and code readability.

You should implement your code in one file (e.g. ex1.cpp or ex1.java). Make sure your program has a header comment block containing the name of the exercise, your name and your student login (e.g. jfk01).

For this exercise you are to write a program (in C, C++ or java, preferably) that reads a text file containing a number of words and displays the words on the screen in reverse order using a stack. A pseudo-code outline for the program is given below:

```
Begin main
    display a prompt for the file name
    read in the file name
    try to open the file
    if the file fails to open
        print an error message on the screen and exit
    fi
do
    read in a word from the file
    if the file read fails
        terminate (break) the loop
    fi
    Push the word onto the stack
od
close the file
while the stack is not empty
    display the top stack word on the screen followed by a space
    pop the top value from the stack
elihw
End main
```

Do not implement the stack using a separate struct or class or with STL. The stack must be implemented using a fixed size array of words and an index integer for indicating the top of the stack. The stack array and index should be global variables (or private in java). A word can be a string or a c-string (i.e. a character array). You can assume no word is more than 20 characters long. The stack functions (i.e. push(), top(), pop(), isEmpty()) should be implemented below the main() and prototyped above the main() in C or C++.

When you are finished, test your program using the provided text file named “ex1.txt” and show your code, the output and the submit receipt to your lab tutor to receive your marks.

To submit your work upload your file to banshee and submit it using the submit instruction below:

```
$ submit -u login -c CSCI203 -a ex1 filename
```

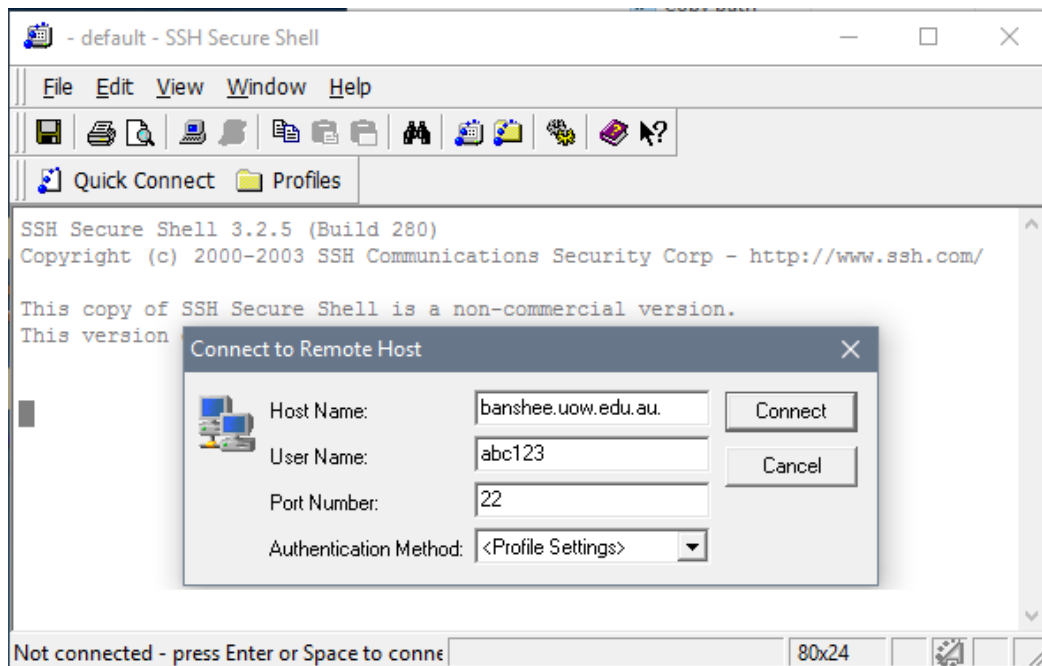
where 'login' is your UNIX login ID and 'filename' is the name of your file.

If you are unfamiliar with remote access to banshee via SSH, see following pages.

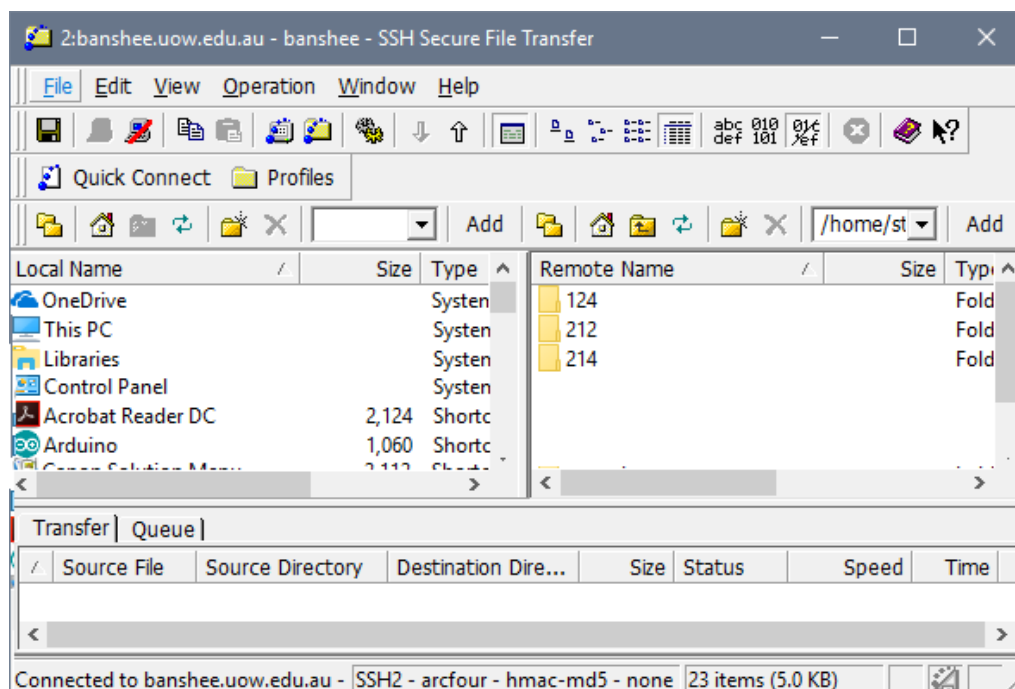
Note: If you are unable to attend your lab class and demonstrate your work on time due to circumstances beyond your control (e.g. sickness), contact your lecturer to request an extension.

How to submit your work on Windows

Start SSH and click “Quick Connect” on the toolbar. Then enter “banshee.uow.edu.au” and your user name in the input dialog as shown below. Also, enter your password when prompted to do so.



Click the “New File Transfer Window” tool on the toolbar to open a file transfer window.



Upload your exercise file (e.g. ex1.cpp) to your unix account by dragging and dropping it onto the remote host side of the file transfer window, then close the file transfer window.

Enter the submit command provided at the end of the exercise 1 specs in the SSH client terminal window and then enter your password in response to the prompt.



```
1:banshee.uow.edu.au - banshee - SSH Secure Shell
File Edit View Window Help
SSH Secure Shell 3.2.5 (Build 280)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.

Last login: Mon Apr  3 18:22:55 2017 from 46.53.240.116.s
$ submit -u abc123 -c CSCI203 -a1 main.cpp ass1.h ass1.cpp output.txt
```

Connected to banshee.uow.edu.au SSH2 - arcfour - hmac-md5 - none 80x24

Check the emailed submission receipt. If any compile errors are present, fix them and resubmit.

Note: Both CSCI203 and CSCI803 should submit to the same submit folder (-c CSCI203).

Show the emailed receipt to your tutor along with your work in Week 2 to receive 2 marks.