CS1020: Data Structures and Algorithm I
Intro Workshop: Developing Java Programs on sunfire

Welcome to CS1020! This is a self-guided document to help you complete your Lab #0 (which will be released on CodeCrunch and announced). It will guide you on the following topics:

- Connecting to the sunfire server
- Editing Java programs
- Compiling and running Java programs

Overview

To compile and run a Java program on the sunfire machine, you need to go through the following steps:

1. Log in to your UNIX account on the **sunfire** machine
2. Create and modify Java programs using **vim** editor
3. Compile and execute Java programs

Please check that you have the following before proceeding:

- Your UNIX account (also called SoC account) on the sunfire server. You can create/re-enable your account by going to this website (use your NUSNET-id and password to log into this website):

  https://mysoc.nus.edu.sg/~newacct

  Note that your UNIX account is a different account from your NUSNET account. Do not mix up the two.

- An SSH Client. If you are using Windows, one option for this is the SSH Secure Shell software. You can find a copy of the software in CS1020 module website, under "Resources", "Online":

  http://www.comp.nus.edu.sg/~cs1020/2_resources/online.html

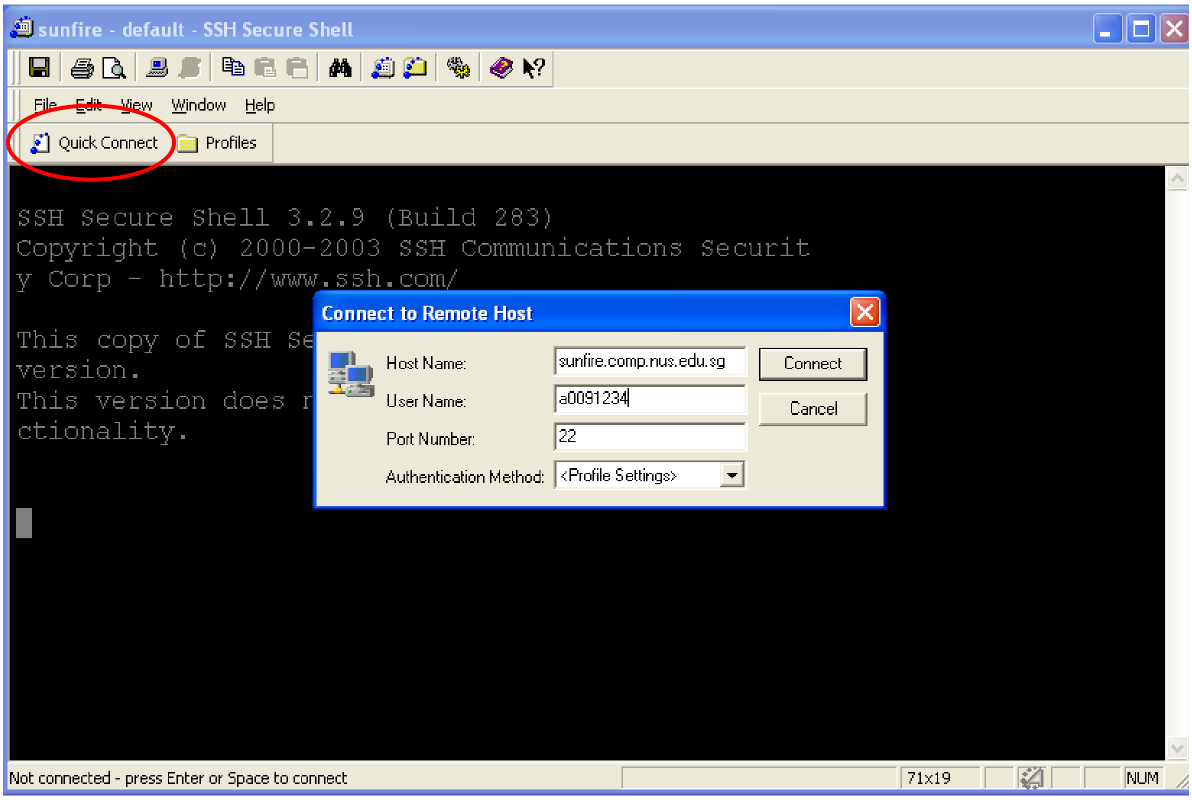
  Please download and install it in your own PC if you intend to access your UNIX account from home.

Those of you working on Macs or Linux variants can use the "ssh" and "scp" commands in your "Terminal.app" window. The instructions in this document refer primarily to the Windows environment.

Section 1: Logging into sunfire

All PCs in the labs are installed with SSH Secure Shell. You should be able to find its icon SSH Secure Shell Client on the desktop. Double-click to launch it. (If you can’t find it on the desktop, left mouse click on the **Start** button start at the bottom-left corner, select **All Programs**, look for **SSH Secure Shell** folder, and left click on **Secure Shell Client**.)

A screen similar to the one shown below will appear. Click on the **Quick Connect** button and a pop-up window **Connect to Remote Host** will appear.



sunfire - default - SSH Secure Shell

```
SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Securit
y Corp - http://www.ssh.com/

This copy of SSH Se
version.
This version does n
ctionality.
```

Connect to Remote Host

Host Name: sunfire.comp.nus.edu.sg

User Name: a0091234

Port Number: 22

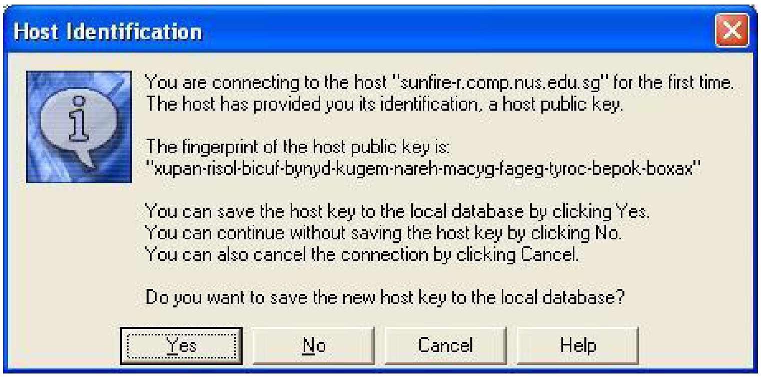
Authentication Method: <Profile Settings>

Not connected - press Enter or Space to connect

71x19

Under Host Name, enter **sunfire.comp.nus.edu.sg** (or simply **sunfire** if you are using a PC in the SoC lab). For your User Name, enter your **UNIX-id**. Note that it is **case-sensitive**. Ensure that the Port Number is 22. Lastly, click on the **Connect** button.

If you encounter the following window, click “Yes” if you don’t want it to appear again next time you log in.



Host Identification

You are connecting to the host "sunfire-r.comp.nus.edu.sg" for the first time. The host has provided you its identification, a host public key.

The fingerprint of the host public key is:
"xupan-risol-bicuf-bynyd-kugem-nareh-macyg-fageg-tyroc-bepok-boxax"

You can save the host key to the local database by clicking Yes.
You can continue without saving the host key by clicking No.
You can also cancel the connection by clicking Cancel.

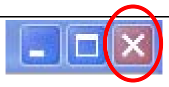
Do you want to save the new host key to the local database?

If the connection doesn't work, you need to install and connect to SoC first using SoC VPN (refer to https://docs.comp.nus.edu.sg/node/1553).

Once the connection is done, another pop-up window will appear prompting you for your **password**. Like your UNIX-id, the password is **case-sensitive** as well (i.e. an "a" is different from an "A"). The password will appear on the screen as asterisks (***) so you need to type slowly and carefully.

Once you have logged into your UNIX account on sunfire, you are automatically placed in your home directory. A directory is similar to a folder in Windows.

Notes before you proceed

- Do not click on the Close button (the button with a cross)  at the top-right corner of the SSH Secure Shell window. The correct way to log out of your UNIX account is explained in Section 8.
- The UNIX system is not GUI-based so the mouse is seldom used (if at all).
- You should see the UNIX prompt (read Section 5). If you don't, check that you have not scrolled off the screen with the mouse. Bring the prompt back into view by scrolling back.
- If you do see the UNIX prompt but whatever you type does not appear, you might have accidentally locked your screen (locking can be done by pressing **control-s** or by pressing the "Scroll Lock" key). Unlock your screen by pressing **control-q** (if you have pressed **control-s** earlier), or by pressing "Scroll Lock" again (if you have pressed "Scroll Lock" earlier).
- If you get stuck after typing a command or in a middle of an application, sometimes typing 'q' (quit) may get you out.
- If typing 'q' does not work, press **control-c** which is the interrupt key in UNIX. Use this as the last resort. If it still doesn't work, seek help.
- Do NOT type **control-z**. (This is a common mistake made especially by DOS users as **control-z** is the interrupt key in DOS.) In UNIX, typing **control-z** sends your job to the background. It appears as if the job has been killed, but it is actually still running in the background. You will get a "There are stopped jobs" message when you try to log out from your UNIX account with jobs running in the background. If this happens, type 'fg' (foreground) to bring the job back to the foreground. Not doing this properly may result in problem later. For example, if you accidentally entered **control-z** in vim, the next time you try to edit the same file in vim, you will get a "Swap file already exists!" message.
- Tip 1: You may use the up and down arrows to select a previously entered command. This saves you some typing time.
- Tip 2: You may use the tab key to fill out the complete filename. Ask your lab TA to show you how this can be done.

Section 2: Password

Always **remember your password**! If you forget, you would need to approach the personnel at the Help Desk (COM1 building, Basement) to submit a request for a new password. Alternatively, you can go to the following website to reset your password online:

https://mysoc.nus.edu.sg/~myacct/

Section 3: Setting up your account

If you are new to the sunfire server, you are encouraged to perform the following setup program. It copies some setup files, such as **.bash_profile** and **.vimrc**, into your home directory. It also copies the **HelloWorld.java** program into your **java** subdirectory. You need only to do this once.

Run the **setup** program as follows. Type the following commands in your SSH window:

```
~cs1020/workshop/setup
```

Check that you enter the above command ACCURATELY (the tilde character ~ is usually found on the top left of your keyboard next to the "1" key).

After the setup program runs successfully, you must type:

```
source .bash_profile
```

Remember to type the SPACE in between "source" and ".bash_profile". Note that if the above command ran successfully, you will see no output. With many UNIX commands, "*no news is good news*". This is very different from Windows or Mac where you are used to seeing pop-up messages or dialog boxes. Try to get used to UNIX quickly.

Note to students who took CS1010 last semester.

If you have already done the setup last semester when you took CS1010, and you do not wish to overwrite your **.vimrc** file or other system files which you may have already customized, you may simply copy the **HelloWorld.java** program into your UNIX account as follows.

First, in your home directory, create a **java** subdirectory (if you do not have a **java** subdirectory yet):

```
mkdir java
```

Copy the **HelloWorld.java** into your **java** subdirectory:

```
cp ~cs1020/workshop/HelloWorld.java java
```

Section 4: Getting Online Help

An online help facility is available in the system via the "`man`" command ("`man`" stands for **MAN**ual). Type "`man man`" (i.e. get the manual page on the command "`man`") to find out more about the facility. To look for more information about any UNIX command, for example, "`ls`", type "`man ls`". To exit "`man`" pages, press "`q`".

Section 5: UNIX Crash Course

The following are a few commonly-used commands. This list is by no means exhaustive and you are urged to explore on your own. Note that UNIX commands are case-sensitive.

In the examples below, bold words are commands which you are expected to enter. Assume that you are `a0091234`, your default prompt may look like this (possibly a little different):

```
a0091234@sunfire0:~[xxx]$
```

`sunfire0` is the internal name for sunfire; ~ indicates that you are currently in your home directory. `[xxx]` indicates the command number in the command history.

a. Directory commands

pwd to **P**rint current **W**orking **D**irectory to show you which directory you are currently in

```
a0091234@sunfire0:~[xxx]$ pwd
/home/a/a0091234
```

ls to **L**i**S**t files in your current directory

```
a0091234@sunfire0:~[xxx]$ ls
java
```

You may also use "`ls -F`" for more information (`-F` is one of the many options/flags available for the `ls` command. To see a complete list of the options, refer to the man pages, ie. "`man ls`".)

```
a0091234@sunfire0:~[xxx]$ ls -F
java/
```

The slash (/) beside the filename tells you that the file is a directory (folder). A normal file does not have a slash (/) beside its name when "`ls -F`" is used.

Note that the directory java is created by the setup program in section 3. If you did not run the setup program, you will not see any file at all.

You may also use the "`ls -l`" command (dash L) to display almost all the file information, include the size of the file and the date of modification. Try it now!

cd to **C**hange **D**irectory from current directory to another

```
a0091234@sunfire0:~[xxx]$ cd java
a0091234@sunfire0:~/java[xxx]$ ls -F
HelloWorld.java
```

Note that the prompt changes to ~/java to indicate that you are now in the `java` directory below your HOME directory.
Entering "`cd`" alone brings you back to your HOME directory, i.e. the directory in which you started with when you first logged into the system.

```
a0091234@sunfire0:~/java[xxx]$ cd
a0091234@sunfire0:~[xxx]$
```

mkdir to **M**a**K**e a sub**DIR**ectory in current directory

```
a0091234@sunfire0:~[xxx]$ mkdir another
a0091234@sunfire0:~[xxx]$ ls -F
another/   java/
```

rmdir to **R**e**M**ove a sub**DIR**ectory in current directory – note that a directory must be empty before it can be removed.

```
a0091234@sunfire0:~[xxx]$ rmdir another
a0091234@sunfire0:~[xxx]$ ls -F
java/
```

b. File commands

cp to **C**o**P**y files

```
a0091234@sunfire0:~[xxx]$ cd java
a0091234@sunfire0:~/java[xxx]$ cp HelloWorld.java anotherfile.java
a0091234@sunfire0:~/java[xxx]$ ls
anotherfile.java    HelloWorld.java
```

mv to **M**o**V**e files from one directory to another; can also be used to rename files.

```
a0091234@sunfire0:~/java[xxx]$ mv anotherfile.java afile
a0091234@sunfire0:~/java[xxx]$ ls
afile    HelloWorld.java
```

rm to **R**e**M**ove files. Be **careful** with this command – files deleted cannot be restored (unless they have been backed up during the normal backup cycle).

```
a0091234@sunfire0:~/doc[xxx]$ rm afile
rm: remove 'afile'? y
a0091234@sunfire0:~/java[xxx]$ ls
HelloWorld.java
```

c. Command to display text files

cat to string together or display (**CAT**enate) the contents of files onto the screen

```
a0091234@sunfire0:~/java[xxx]$ cat HelloWorld.java
```

less variant of "`cat`" (includes features to read each page leisurely)

```
a0091234@sunfire0:~/java[xxx]$ less -e HelloWorld.java
```

Preferred for larger files. In "`less`", use <space> to move down one page, 'b' to move **B**ack up one page, and 'q' to **Q**uit from "`less`". You can also use the up/down arrow keys to move one line at a time.

Section 6: Running a Java Program

The process of creating a working Java program involves the following steps:

```
        +---------+
+---+-->|  EDIT   |
|   ^   +---------+
|   |        |
|   |        |
|   |        v
|   |   +---------+
|   +---| COMPILE |
|       +---------+
|            |
|            |
|            v
|       +---------+
+-------|   RUN   |
        +---------+
```

1. Use an editor of your choice, type in the source code (filename must have extension `.java`, eg: `HelloWorld.java`)
2. Compile the source code to obtain the bytecode. Command is **javac**. If there are compilation errors, identify them and re-edit the source code before you proceed.
3. Run the bytecode. Command is **java**. If there are run-time errors, you must identify them and re-edit the source code, and compile again.

Section 7: Creating your own Java Programs

Create your first Java program "HelloWorld.java"

There are a number of editors available in sunfire: *vim*, *vi*, *emacs*, *joe*, *pico*, *nano* etc. The more powerful one requires a longer time to learn. For this lab, you will use **`vim`** – a powerful editor with many commands, but even with the knowledge of a few simple commands it is quite easy to use and very powerful. It is YOUR responsibility to pick an editor and master it, and in future labs we will assume that you are familiar with your editor and its various functions.

It is recommended that you create a directory to store all your Java programs. Placing all programs under your home directory can get messy real quick. If you have done the setup in section 3, the java subdirectory would have been created. Otherwise, perform the following steps to create a new subdirectory:

1. Enter "**`cd`**" (this will get you back to the home directory)
2. Enter "**`mkdir java`**" (creates a new directory java under home directory)
3. Enter "**`ls –F`**"
   You should see the new directory **`java`** along with other files and directories.

Go to the **`java`** directory. Enter "`vim HelloWorld.java`", then press the "`i`" key. You will see that the word "`-- INSERT --`" appears at the bottom left corner of your screen. You are now in `INSERT` mode. While you are in `INSERT` mode, you may use the arrow keys (Up, Down, Left, Right) to move around your program, as well as the Backspace key and Delete key to delete text. You can use the arrow keys in Insert mode after you set up your `.vimrc` profile. The PgUp and PgDn keys do not work, so do not use them. Do not use the scroll bar as it does not always work as expected.

Notice that line numbers (1, 2, 3, etc.) are displayed on the left. This helps you to easily identify a number by its line. Line numbers are NOT part of the actual Java code that you write, but are provided by *vim* to assist you in coding. This is extremely useful when identifying the location of compilation errors.

To copy text, you may use the mouse to highlight blocks of text, then right-click on the mouse and choose Copy. To paste your text, you must use the cursor keys to move the cursor to the desired position, then right-click on the mouse and choose Paste.

Type in the following program:

```
import java.lang.*;

public class HelloWorld {

   public static void main(String[] args) {

       System.out.println("Hello World!");

   }
}
```

Note that for simplicity, the above program has no documentation. A good program should include documentation, at least the identity of the author, the purpose of the program and other relevant information. Keep this in mind when you write your own programs.

When you are done, press `<ESC>` (to get into `COMMAND` mode), then type `ZZ` (capital `Z` twice). You may also press "`<ESC>:x<ENTER>`" (more clearly seen as pressing the following four keys one after another: `<ESC> : x <ENTER>`) to save your program and exit from the *vim* editor. `<ESC>` means press the Escape key, then press the colon key (shift-;), then press `x` (the `x` is a lowercase `x`), and finally `<ENTER>` means press the Enter key.

If you want to save your file without exiting from the *vim* editor, press "`<ESC>:w<ENTER>`", then press the "`i`" key again to go back into `INSERT` mode. It is a good habit to save your file periodically so that if the network or the system goes down for any reason, you will not lose your hard work.

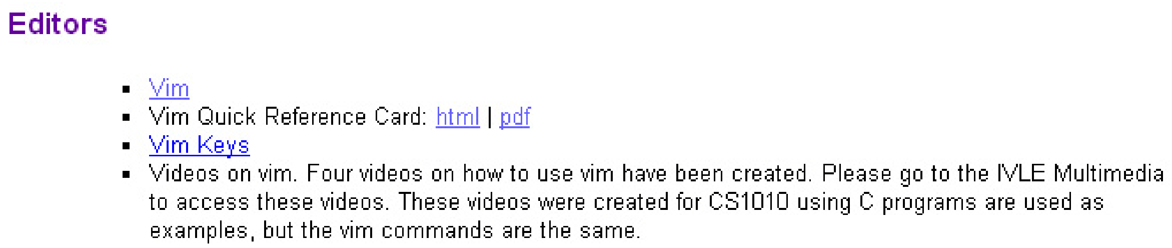
When you startup *vim*, it begins in `COMMAND` mode. One way to go into `INSERT` mode is to press the "`i`" key. While we are in `INSERT` mode, we can type in our Java code. To switch back to `COMMAND` mode, we press the "`<ESC>`" key.

The following shows you some commands on how to save your file and exit in vim or quit from *vim*:

- `<ESC>`**`:wq`**`<ENTER>` : Saves your program and exits from *vim*.
- `<ESC>`**`ZZ`** (Note that the Zs are uppercase) Saves your program and exits from *vim*.
- `<ESC>`**`:q!`**`<ENTER>` : Quits from *vim* without saving your program. All changes done since the last save will be lost.

To find out more about *vim* commands, go to the module website, "Resources", "Online" page:

http://www.comp.nus.edu.sg/~cs1020/2_resources/online.html



Editors

- Vim
- Vim Quick Reference Card: html | pdf
- Vim Keys
- Videos on vim. Four videos on how to use vim have been created. Please go to the IVLE Multimedia to access these videos. These videos were created for CS1010 using C programs are used as examples, but the vim commands are the same.

Compiling and running your program

```
a0091234@sunfire0:~/java[xxx]$ javac HelloWorld.java
```

There will be error messages if your program has errors. Edit your program to make the necessary corrections and re-compile. If there are no compilation errors, a class file **HelloWorld.class** will be created.

Proceed with program execution as follows:

```
a0091234@sunfire0:~/java[xxx]$ java HelloWorld
```

If you feel comfortable with the above steps, you can try out some bigger Java programs from the textbook. Do not be discouraged by the compilation errors (you are bound to get a few ☺). It is more important to understand compilation errors quickly: what they mean and how to fix them.

Opening multiple sunfire windows

It might be advisable to open multiple sunfire windows for the convenience of development: one window for coding (editing your program), and another window for running/testing/debugging.

Section 8: Logging Out

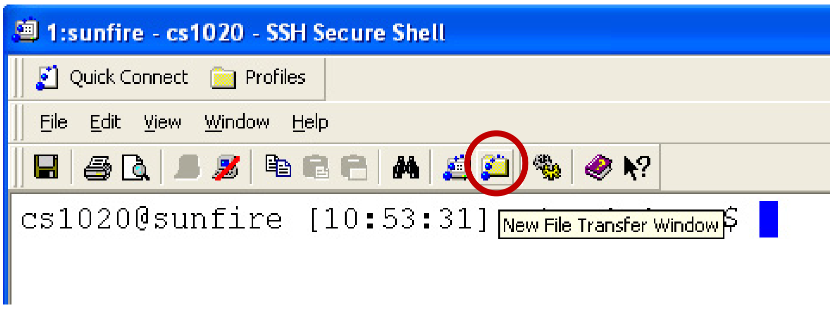
When you have finished, you must always log out of the computer system as follows:

```
a0091234@sunfire0:~/java[xxx]$ logout
```

Typing "`exit`" (or pressing **control-d**) also has the same effect. You MUST remember to log out, otherwise anyone who comes by may delete all your valuable files away!

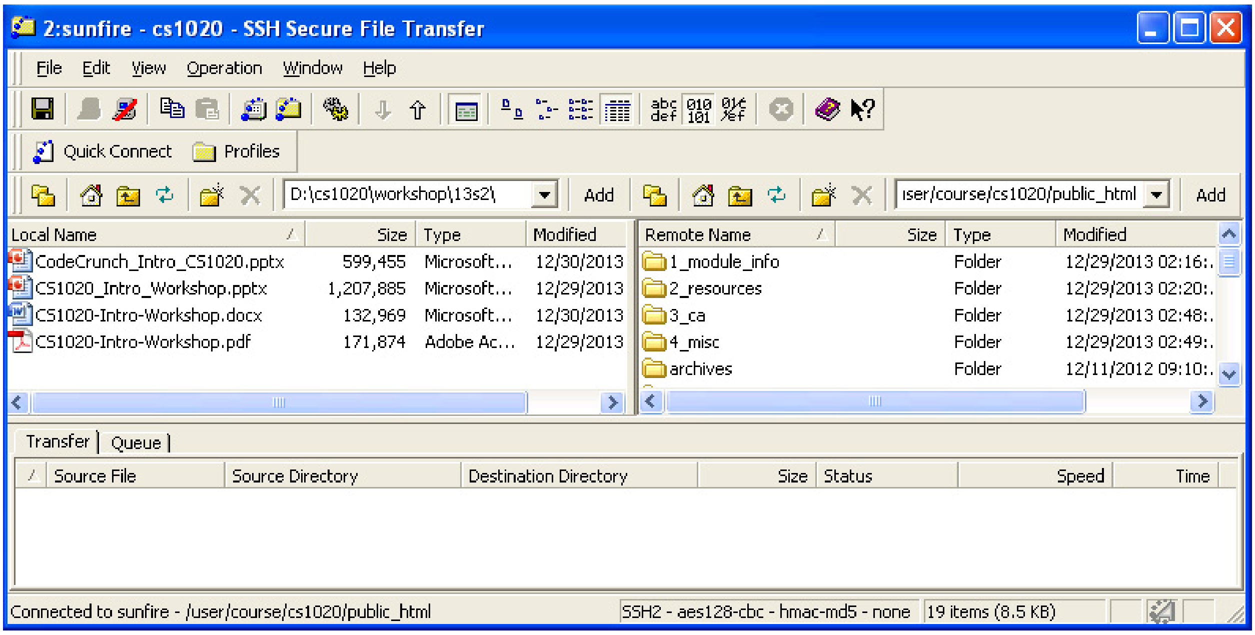
Appendix: Transferring files to and from sunfire

The easiest way to transfer files between your sunfire account and your local machine/host/PC/laptop is to login to sunfire first using Secure Shell Client, then open the File Transfer program by clicking on File Transfer icon as circled in the diagram below.



```
cs1020@sunfire [10:53:31] $
```

You will see the files in your local machine on the left side and the files in your sunfire account on the right. You may navigate around and copy and delete files similar to how you would do it on Windows.



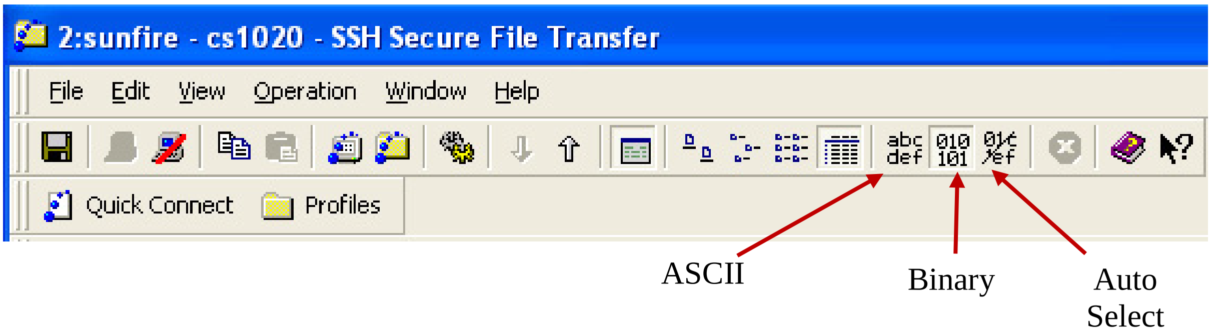
Local Name	Size	Type	Modified
CodeCrunch_Intro_CS1020.pptx	599,455	Microsoft...	12/30/2013
CS1020_Intro_Workshop.pptx	1,207,885	Microsoft...	12/29/2013
CS1020-Intro-Workshop.docx	132,969	Microsoft...	12/30/2013
CS1020-Intro-Workshop.pdf	171,874	Adobe Ac...	12/29/2013

Remote Name	Size	Type	Modified
1_module_info		Folder	12/29/2013 02:16:.
2_resources		Folder	12/29/2013 02:20:.
3_ca		Folder	12/29/2013 02:48:.
4_misc		Folder	12/29/2013 02:49:.
archives		Folder	12/11/2012 09:10:.

Source File	Source Directory	Destination Directory	Size	Status	Speed	Time

Connected to sunfire - /user/course/cs1020/public_html | SSH2 - aes128-cbc - hmac-md5 - none | 19 items (8.5 KB)

To transfer a file or files, select the transfer mode: ASCII (for text files), Binary (for binary files) or Auto-Select (useful if you are transferring many files at once or a directory of files) first. Then select and drag the file(s)/directory you wish to transfer from the source to the destination.


2:sunfire - cs1020 - SSH Secure File Transfer
File
Edit
View
Operation
Window
Help
Quick Connect
Profiles
ASCII
Binary
Auto
Select