

2

UNIX SAS

For readers who have no access to expensive SAS subscription, please start from next chapter, Chapter 3: Free SAS University Edition. For this book, readers can use the free version, to finish 80% of the tasks. In other words, students at schools without a valid SAS subscription could still learn SAS.

Generally speaking, there are two types of SAS: PC SAS/or Mac SAS which is run on individual users' machines, and UNIX SAS which is run on a server. There are many advantages of running a PC SA such as it is 100% under your control, colorful SAS editor, and good for producing all types of graphs. On the other hand, there are many advantages of using UNIX SAS. The one of the major advantages of using UNIX SAS is that users could deal many commonly used data sets. Another advantage is that researchers could access a server from anywhere. The only tool researchers need is to access internet. In particular, the following topics will be covered:

- How to connect to a server
- SSH
- Putty and ftp
- MobaXterm
- Several widely used UNIX/Linux commands
- UNIX text editor vi
- Running UNIX SAS
- Running UNIX command from SAS
- Serverless

2.1 PROS AND CONS OF USING UNIX SAS

Compared with PC SAS, UNIX SAS has its own advantages and disadvantages. Let's talk about the advantages first. One of the most important advantages is that users do not need

SAS on his/her machine. The following Table lists most advantages of using UNIX SAS, Stanford (2004). This means that the only thing, a researcher or user need to have, is a computer with internet access.

Table 2.1: Advantages of using UNIX

#	Description
1	Researchers and users don't need SAS installed on his/her computers
2	Researchers and users could share many common data sets. This is especially true for most used financial/economics/accounting databases
3	Full multitasking with protected memory, users can run multiple programs at the same time
4	Efficient virtual memory, many programs can run with a modest amount of physical memory
5	Access controls and security. Files are owned by particular accounts.
6	A rich set of small commands and utilities that do specific tasks well -- not cluttered up with lots of special options.
7	Ability to string commands and utilities together in unlimited ways to accomplish more complicated tasks -- not limited to preconfigured combinations or menus, as in personal computer systems.
8	A powerfully unified file system. Everything is a file: data, programs, and all physical devices. Entire file system appears as a single large tree of nested directories, regardless of how many different physical devices (disks) are included.
9	A lean kernel that does the basics for you but doesn't get in the way when you try to do the unusual
10	Available on a wide variety of machines - the most truly portable operating system.
11	Optimized for program development, and thus for the unusual circumstances that are the rule in research

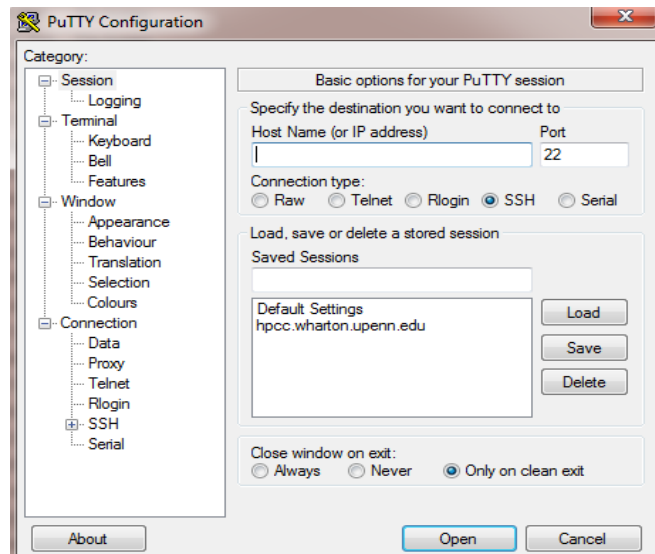
The disadvantages are listed below. For the table, some entries are borrowed from Stanford (2004).

Table 2.2 Disadvantages of using UNIX

#	Descriptions
1	Users need to learn some UNIX commands
2	Users need to learn one UNIX text editor (for writing programs)
3	The traditional command line shell interface is user hostile -- designed for the programmer, not the casual user.
4	Commands often have cryptic names and give very little response to tell the user what they are doing. Much use of special keyboard characters - little typos have unexpected results.
5	To use Unix well, you need to understand some of the main design features. Its power comes from knowing how to make commands and programs interact with each other, not just from treating each as a fixed black box.
6	Richness of utilities (over 400 standard ones) often overwhelms novices. Documentation is short on examples and tutorials to help you figure out how to <i>use</i> the many tools provided to accomplish various kinds of tasks.

2.2 USING SSH TO CONNECT TO A SERVER

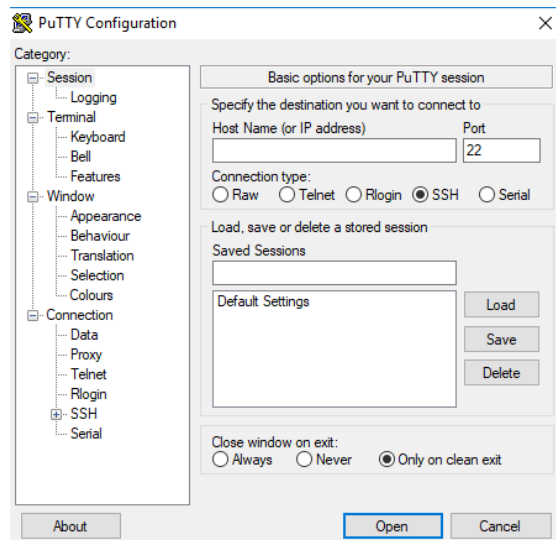
Before we can run a UNIX SAS program, we have two extra tasks: install a tool to connect to our server and learn one text editor to write and editor SAS programs. There are several ways to connect to a UNIX server. The easiest way is to use putty.exe software which can be downloaded at <https://www.putty.org/>. First, after launching Putty, the following window would pop up.



Enter your server name in the box called “Host Name (or IP address)”. After click Open, we would enter our user name and password. If nothing is wrong, we would be connected with the specified server. Two more methods to connect with a server: via SSH (see Appendix B) and via MobaXterm (see Appendix C).

2.3 USING PUTTY TO CONNECT TO A SERVER

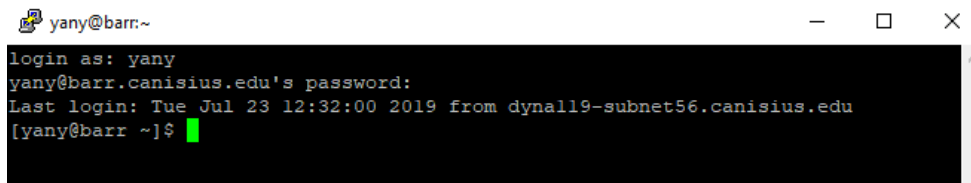
First, reader should download Putty.exe at <https://putty.org/>. After clicking putty.exe, we could see the following image.



At the “Host Name (or ID address), readers should enter the correct location. Assume that we enter barr.canisius.edu. After clicking Open, we would see the following image

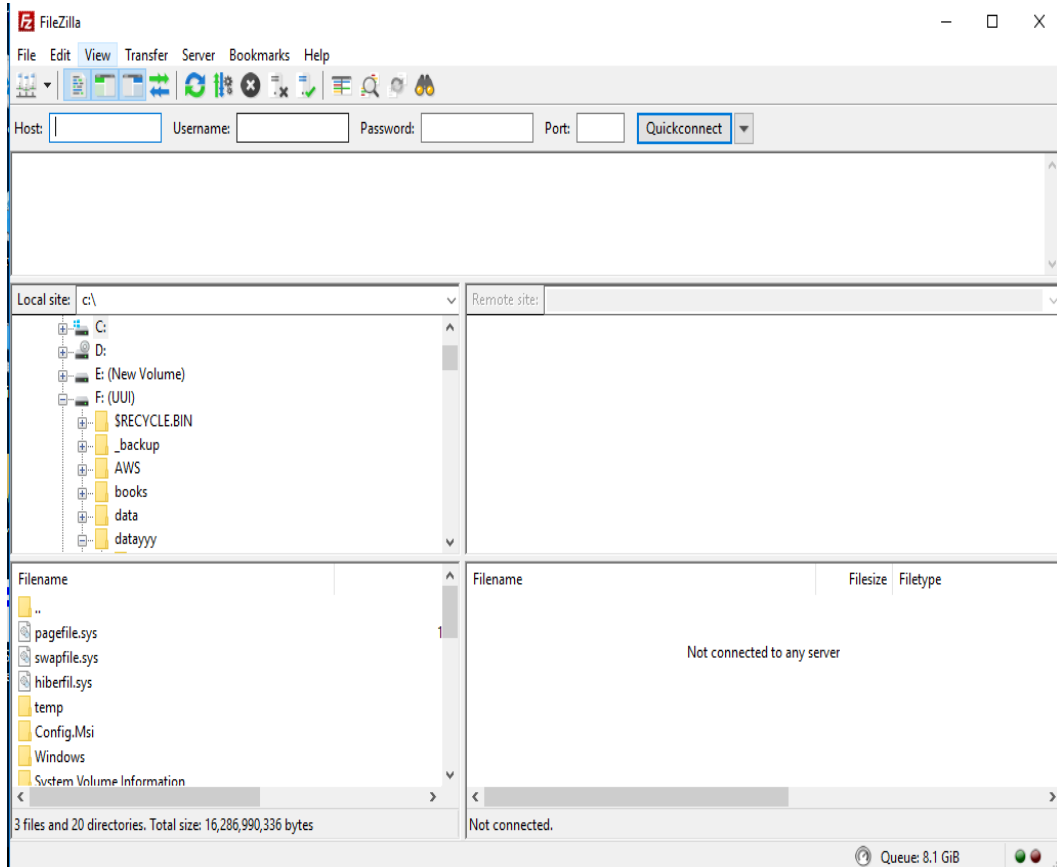


Enter your username and password, we could have the following result.



2.4 FILEZILLA TO UPLOAD AND DOWNLOAD FILES

We can use FileZilla to transfer files (upload and download) files. First, readers should download the software at <https://filezilla-project.org/download.php>. After clicking the file, we could see the following image.



2.5 SEVERAL UNIX COMMANDS

When dealing with a UNIX/Linux server, we need to learn a few UNIX commands in order to move around, generate a subdirectory, copy and delete files and the like. The following table shows several most widely used UNIX commands. Note that for UNIX/Linux servers, they are case sensitive.

Table 2.3 widely used UNIX commands

<code>ls</code>	list files in current directory
<code>ls -l</code>	long list of files in current directory
<code>ls -a</code>	list all files (including hidden) in current directory
<code>cd research</code>	moves into the directory with a name of research
<code>cd</code>	cd on its own takes you back to your home directory
<code>cd ..</code>	moves up one level in directory hierarchy
<code>mkdir teaching</code>	makes a new directory with a name called `teaching'
<code>rmdir research</code>	removes the (empty) directory with a name called `research'
<code>rmdir -r dirname</code>	recursively removes directories and subdirectories
<code>cp file1 file2</code>	makes a copy of file1 and calls it file2
<code>rm file1</code>	removes (deletes) file1
<code>rm -i file1</code>	asks for confirmation that you want to delete file1
<code>vi t.sas</code>	Start a UNIX text editor called vi to open t.sas program
<code>head t.sas</code>	Show the first several lines of t.sas
<code>tail t.sas</code>	Show the last few lines of t.sas
<code>cat t.sas</code>	Type t.sas on the screen
<code>cat t.sas grep ret</code>	Catch phrase of "ret" from t.sas program
<code>man ls</code>	Find instruction (manual) for ls command

2.6 VI: A UNIX TEXT EDITOR

After we log on to a server, we need an editor or UNIX editor to write and edit our programs. There are several such editors such as : vi, pico, and emacs. In this section, we focus on vi. First we start with how to launch the vi editor and exit it.

Table 2.4 A list of important commands for vi editor

Command	Meaning
vi t.txt	will open a file called t.txt
ZZ	exit vi, saving changes
:q!	quit, discarding changes

For vi editor, there exist two modes: Command mode and editing mode

Action	Meaning
[escape]	Command mode
i	(lower case letter i) Input mode

On Command mode: Movement of the cursor

Command	Meaning
h	Move the cursor to the left
j	Move the curse down
k	Move up
l	Move right
20 j	Move 20 lines below
G	Go to the end of the file
1G	Go to the beginning of the file
ESC	End editing mode, back to command mode

Delete a character or a line or several lines.

Command	Meaning
x	Delete the current character
dd	Delete the current lines

3dd	Delete 3 lines starting from the current one
u	Undo the previous change
/ret	Search for text call 'ret'
n	Repeat the search
ctr-g	Show the current location to total number of lines
ESC	End editing mode, back to command mode

Copy one line or several lines, then paste them.

Command	Meaning
y	Copy the current line
P	Past the copied line at the current position
3y	Copy 3 lines

Retrieve a file to your current file. Assume that you have a file called header.sas with the following contents. Note on command mode, the command starts with a ':' and 'r' stand for retrieve.

Command	Meaning
:r header.sas	Retrieve file called header.sas to the current file

Here are the contents of the file called header.sas which defined the options for your output file. In other words, the output contains no date, not cantered, no number. PS refers to page size and ls for length size. The 'fullstimer' asks the system output all system information to the log file.

```
options nodate nocenter nonumber ps=max ls=72 fullstimer;
```

2.7 RUNNING A UNIX SAS PROGRAM

When running UNIX SAS, two extra files will be generated with the same name but different extensions. For example, if running a program called t.sas, t.lst (output file, called print file) and t.log (log file) will be generated. The original SAS data set is called t.sas and we use the following code to run it.

```
$sas t.sas &
```

Depending on system, the above code might be a little different. For example, for some systems, we could have the following one called runSAS.sh.

```
#!/bin/bash
nohup sas -nodisplay -nodesktop -nosplash t.sas
```

Then we issue the following code.

```
qsub t.sh
```

The program itself is shown in the next section. The extra files with the same program name but different extension would be generated. The file called `t.log` is the log file which contains information about our program, while `t.lst` is the output or listing file which contains all output.

```
-rw-rw-r-- 1 ext-yxyan ext-yxyan 2373 Jun 26 13:59 t.log
-rw-rw-r-- 1 ext-yxyan ext-yxyan  422 Jun 26 13:59 t.lst
-rw-rw-r-- 1 ext-yxyan ext-yxyan   69 Jun 26 13:59 t.sas
```

For UNIX SAS, we issue the following line to run it.

```
$sas t.sas &
```

The ampersand of `&` could be omitted. If there is no ampersand, our cursor would return only after the whole program is finished. As mentioned before, two more files would be generated with different extensions. Since our program is called `t.sas`, files called `t.lst` and `t.log` would be generated. The following image shows the contents of `t.lst`.

```
The SAS System                                13:59 Tuesday, June 26, 2018  1

Obs      x
1      1.0
2      2.3
3      4.5
```

The following image shows the contents of `t.log`.

```
The SAS System

NOTE: Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software 9.4 (TS1M4)
      Licensed to UNIVERSITY OF PENNSYLVANIA - SFA T&R, Site 70055201.
NOTE: This session is executing on the Linux 2.6.32-696.20.1.el6.x86_64 (LIN X64) platform.

NOTE: Updated analytical products:

      SAS/STAT 14.2
      SAS/ETS 14.2
      SAS/OR 14.2
      SAS/IML 14.2
      SAS/QC 14.2

NOTE: Additional host information:

Linux LIN X64 2.6.32-696.20.1.el6.x86_64 #1 SMP Fri Jan 12 15:07:59 EST 2012 x86_64 Red Hat 1
(Santiago)

*** Welcome to Wharton Research Data Services (WRDS) ***

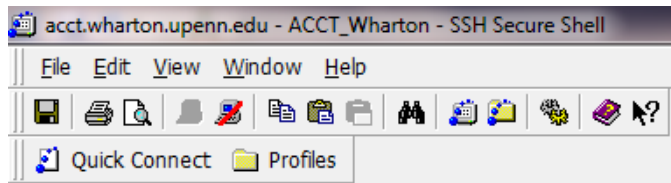
WRDS is developed and run by the Wharton School at the University of
Pennsylvania. We provide access to the databases that define empirical,
seminal research in the fields of accounting, banking, economics,
finance, insurance, management, marketing, public policy, risk
management, and statistics.

If you are not an authorized user of this system, disconnect now.
All activity is strictly monitored. There may be severe criminal and
civil penalties for unauthorized access or use of computing resources.
```

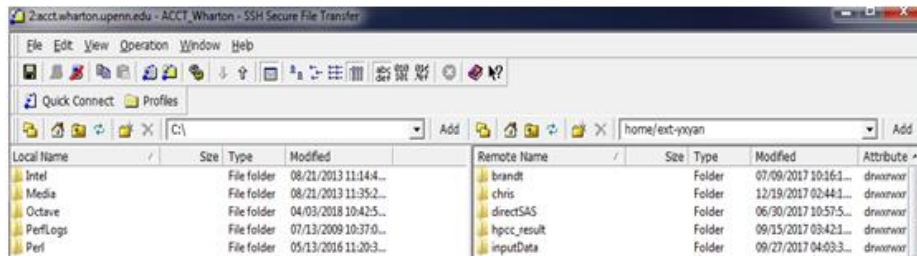
For some UNIX system, we use the following line to run a SAS program.

```
sas -noterminal t.sas &
```

Note that when using UNIX SAS, we could have many tools to connect with a server. One good tool is called SSH (see the following image).



It has a download and upload window, see the following image.



The left panel is for our window or PC while the right panel is for our UNIX system. It is quite convenient to upload programs and our initial data sets and download final results.

2.8 RUNNING UNIX COMMANDS FROM SAS

Occasionally, we might need to run a UNIX command from SAS. For example, we have many zip files. In order to retrieve data from each of them, we might need generate a list file by using the following UNIX command: `unzip -l myzipfile.zip >list.txt`. When our SAS finishes this part and we don't need this list.txt file and could manually remove it by issuing the UNIX command of `rm list.txt`. For one zip file, it is okay that we do it manually before and after running our SAS program. However, if we have n such zip files, it might be a good idea to include such UNIX commands within our loop.

```
x unzip -l myzipfile.zip > list.txt
# our code here
x rm list.txt
```

2.9 HOW TO DEBUG?

In the previous chapter, we mention one method to debug. That method could be applied to both PC SAS and UNIX SAS. For UNIX SAS, we search for ERROR from the log file. For example, we use "ta temp;" for the first line instead of "data temp;". We would find the following error message.

```
1          ta temp;
          —
          180
```

ERROR 180-322: Statement is not valid or it is used out of proper order.

2.10 SAVING MANY SMALL PROGRAMS

There are many ways to learn a programming language. One of the key components is hands-on. It means that students should spend more time writing their program to retrieve, process and run various models. By debugging their own and others' programs, they could learn more. Another good habit is to save their own small programs. We call this "small programs oriented". This means at the end of this course or this book, readers could have up to 100 their own programs. This could offer several benefits. First, during mid-term or final exams, students don't need to start their program from scratch. Instead, they could find a related program, then modify it to satisfy their need. After graduation and landing a job with a company, they still could use those small programs at their will. Since they have tested those programs, they would feel more confident when using them. Over the years, they would add more and more such small programs. For example, one of the author of this book has collected close to 10,000 SAS programs.

The command `dir *date*.sas` could be used to list all SAS program with "date" embedded in their names.

```
C:\yan\Re\Talisma>dir *date*.sas
Volume in drive C is OSDisk
Volume Serial Number is E0D9-40C3

Directory of C:\yan\Re\Talisma

10/08/2007  11:26 AM                2,640  accouning_date_research.sas
10/08/2007  11:26 AM                2,640  accouning_date_within_same_industry.sas
02/04/2010  01:26 PM                2,216  auto_update_fama_french.sas
02/04/2010  02:48 PM                1,245  auto_update_fama_french_good.sas
01/02/2008  02:56 PM                 637  bankscope_consolidated_code.sas
04/01/2003  02:43 PM                2,439  begdate.sas
05/09/2008  10:38 PM                2,792  check_c0704_unique_date.sas
09/19/2008  02:42 PM                 832  combine_date_with_each_permno.sas
10/27/2004  10:32 AM                2,118  compare_st_date_msfnames_with_cstlink_lin
ddt.sas
11/29/2007  05:02 PM                 532  compg_earnings_announcement_date.sas
12/08/2003  02:15 PM                 442  comp_date_and_yeara_different.sas
02/18/2009  02:47 PM                1,107  comp_filter_daadata_vs_fyear.sas
05/15/2009  09:54 AM                2,235  comp ftp_quarterly_report_date_quarterly_
earnings.sas
```

2.11 RUNNING SAS VIA ANACONDA

Note that this section is optional for most readers. This is especially true for users who have no clue about other languages. Anaconda is a platform that could be used to run many

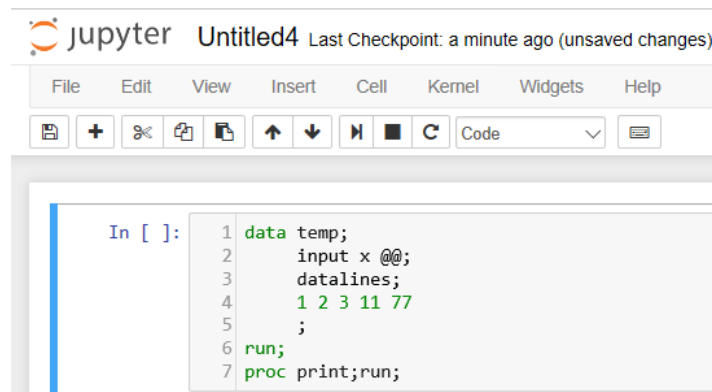
software such as R, Python, Octave, Julia and SAS. As the first step, we have to install a package called sas kernel by issuing the following command.

```
pip install sas_kernel
```

Then, we launch Anaconda.



After clicking SAS (see the above image on the right), we can enter a SAS program, shown below.



2.12 SERVER-LESS

Based on Amazon's statement that Serverless is the native architecture of the cloud that enables researchers or users to shift more of their operational responsibilities to certain server supplied such as AWS (Amazon Web Services). The services increase researchers' agility and innovation. Such services, without own servers, allows researchers to build and run applications and services without thinking about servers and related costs. It eliminates infrastructure management tasks such as server or cluster provisioning, patching, operating system maintenance, and capacity provisioning.

Serverless enables researchers and users to build applications with increased agility and lower total cost of ownership. Building serverless applications means that developers can focus on their core product instead of worrying about managing and operating servers or runtimes, either in the cloud or on-premises. This reduced overhead lets developers reclaim time and energy that can be spent on developing great products which scale and that are reliable, Amazon (2019).

REFERENCES

- FileZilla download, <https://filezilla-project.org/download.php>
- Lin, Chii-Dean, Ming Ji, UNIX Comes to the Rescue: A Comparison between UNIX SAS® and PC SAS®, https://www.lexjansen.com/wuss/2004/coders_corner/c_cc_unix_comes_to_the_rescue.pdf
- MobaXterm download page at <https://mobaxterm.mobatek.net/download.html>
- Parvez, Irshad, 2017, SAS vs R vs Python - Which is the Best Analytics Tool to Learn?, <https://www.linkedin.com/pulse/sas-vs-r-python-which-best-analytics-tool-learn-irshad-parvez/>
- Putty download, <https://putty.org/>
- SAS home, https://www.sas.com/en_us/home.html
- SAS Product Documentation, <https://support.sas.com/documentation/>
- SAS University Edition, https://www.sas.com/en_us/software/university-edition/download-software.html
- Serverless, <https://serverless.com/framework/>
- Serverless, What is Serverless?, 2019, <https://aws.amazon.com/serverless/>
- Stanford University, 2004 , Advantages and disadvantages of Unix, <https://earthsci.stanford.edu/computing/unix/overview/advantages.php>
- SSH Download, <https://www.ssh.com/ssh/putty/download>
- Twenty most used UNIX commands, <http://www-users.york.ac.uk/~hcb1/unix20.html>
- Zhao, James, 2000, Good Programming Practice when Working Across PC SAS and UNIX SAS, <http://support.sas.com/resources/papers/proceedings17/1363-2017.pdf>

SUMMARY

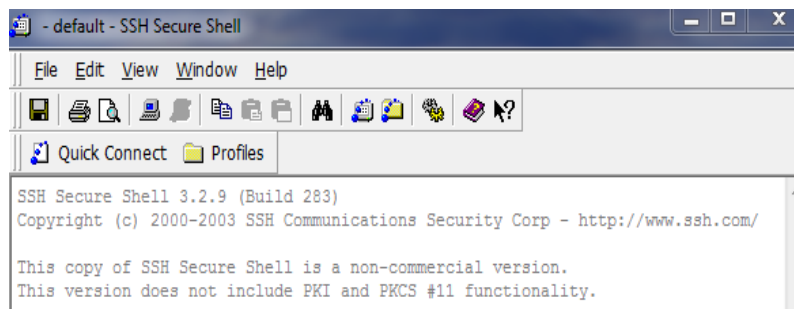
In this chapter, we have discussed how to run UNIX SAS. For readers with a UNIX account, we have mentioned a few UNIX command, how to connect to a UNIX/Linux server and how to run a UNIX SAS program.

For the next chapter, we will discuss free SAS (SAS University Edition). The concept of Virtualbox is very important since many software could be installed on Virtualbox. For this reason, we will discuss how to install Virtualbox. Then we show how to install SAS University Edition into Virtualbox. A few examples will be discussed. For readers who have access expensive SAS subscription, could skip the next chapter. In other words, readers who finish the first two chapters could go to chapter 4 directly.

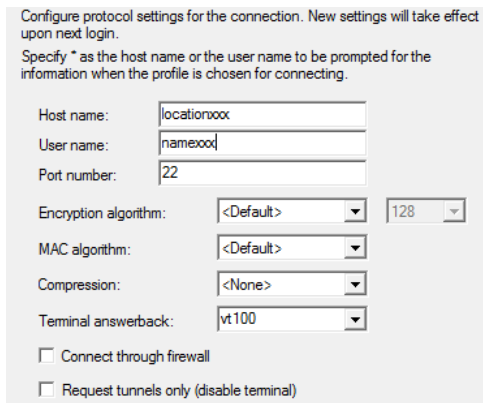
Appendix A: using SSH to connect to a UNIX sever

Download this software at <https://www.ssh.com/ssh/download/>

Step 1: launch SSH, see the image blow



Step 2: Click “Profiles” on the menu bar, click “Add Profiles...”, then click “Edit Profiles...”. Now enter information about your server, see the following image.

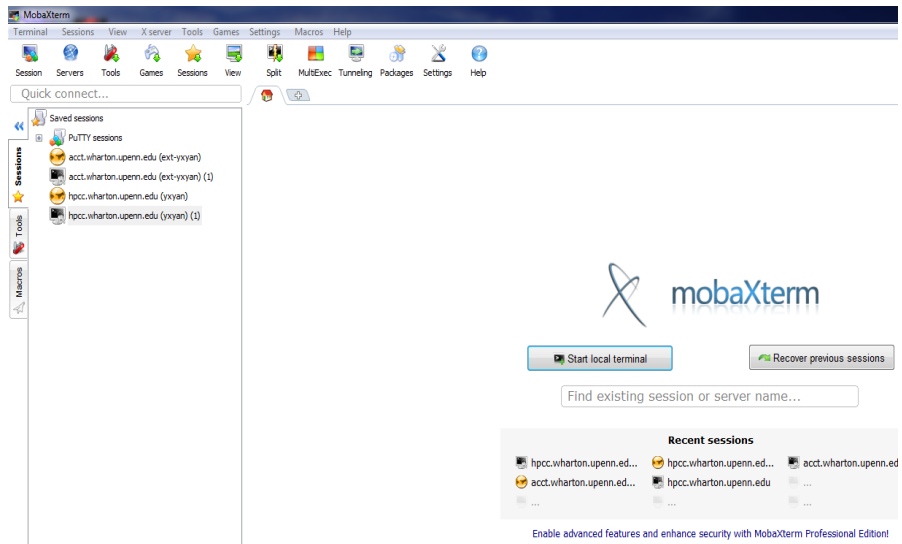


Step 3: we could to use SSH to connect to your serer.

Appendix B: using MobaXterm to connect to a UNIX server

First, we could download MobaXterm at <https://mobaxterm.mobatek.net/download.html>

After we launch the software, we could see the following image.



We could use the software to connect to a specific server.

Appendix C: Command commands for the vi editor for UNIX

1) Open and quit a file	
vi t.txt	Will open a file called t.txt
ZZ	Exit “vi” and save the changes [under “command mode”]
:q!	Quit and discard changes
2) Two modes	
[Esc]	Command mode (Esc is the top left key)
I	(lower case i) enter an Edit mode
3) Under Command mode	
H	(lower case letter of h) Move to the left
J	(lower case letter of j) Move down
K	(lower case letter of k) Move up
L	(lower case letter of l) Move left
X	(lower case letter of x) Delete a character
dd	Delete a line, i.e., delete the current line
U	Undo previous change
/return	Search for text "return"
n	Repeat the research
G	[capital G] Go to the end of the file
1G	[number 1 and capital G] Go to the beginning of the file
ESC	Ends insertion back to command mode
ctr-g	Show current location, # of lines
Y	Copy the current line
P	Paste the current line
3) number before a command	
3dd	Delete 3 lines
3Y	Copy the following 3 lines
3Y ‘then’ p	Copy three lines, the move to another place, hit p to paste those 3 lines

Appendix D: After launching R, just type the following two line of R code.

```
source("http://datayyy.com/sas/week2.R")
.c2

> .c2
function(i) {
" i Chapter 2: UNIX SAS and related issues
- -----
1 Advantages of using UNIX SAS
2 Comparisons between UNIX SAS and PC SAS
3 Server at Canisius College
4 SSH to connect to a server
5 Putty to connect to a server
6 UNIX: connect to a server via SSH
7 UNIX: connect to a server via putty
8 UNIX: a set of UNIX commands
9 UNIX text editor: vi
10 1st SAS program
11 How to run a UNIX SAS program
12 SAS three files
13 Proc print;
14 Debut #1: cat t.log |grep ERR
15 Debut #2: /ERR
16 Running a UNIX command from SAS
17 Server-less
18 AWS (Amazon Web Services)
19 YouTubes
20 Links

Example #1:>.c2 # see the above list
Example #2:>.c2(1) # see the 1st explanation
```

EXERCISES

2.1 What does is mean UNIX server?

- 2.2 What are the advantages of using a UNIX server?
- 2.3 What are the disadvantages of using a UNIX server?
- 2.4 How to connect to a UNIX/Linux server?
- 2.5 Are UNIX/Linux command case sensitive?
- 2.6 If you use UNIX, try the vi editor to write a SAS program and run it.
- 2.7 Generate a directory called teaching and upload a few SAS programs and data sets to that directory.
- 2.8 Use man to find more information about a few UNIX commands such as ls, cd, rm, mkdir
- 2.9 Generate a directory called myData and myPrograms then copy a few files to those two directories. Then remove them.
- 2.10 Using SSH to upload a few files to your UNIX account
- 2.11 Download data sets at <ftp://ftp.nyxdata.com/Historical%20Data%20Samples/>, then use SSH to upload them to your UNIX account.
- 2.12 Try putty.exe to connect to your UNIX/LINUX server and compare it with SSH. What are the advantages and disadvantages of both of them?
- 2.13 What is putty?
- 2.14 How to use putty to connect to a server?
- 2.15 Using Putty to connect to your own server. For Canisius, using barr.canisius.edu.
- 2.16 What does it mean severless?
- 2.17 What are the advantages and disadvantages of serverless?
- 2.18 Who supplied Serverless services?
- 2.19 Compared with SSH, what are advantages and disadvantages of using Putty.
- 2.20 Using vi to write a SAS program and run it on a server.

© Yuxing Yan, 7/2/2020, pyan@geneseo.edu