

Introduction to shell script

Lab 1

[Objective]

1. Download putty and login server.
2. Learn shell command and execution of your first shell script.

[Course Information]

1. **Lab exercise:** The content of lab exercise is prepared according to the knowledge introduced in lectures.
2. **Assignment:** The assignment is independent from the lab exercise and will be sent to you by SAKAI. All of you should submit the assignment before the deadline.
3. **Grading Scheme:**
 - Lecture attendance (10%)
 - Lab attendance and exercise (10%)
 - Assignments (30%)
 - Project (20%)
 - Final exam (30%)
4. **Project:** The details of the project will be announced in the middle of the course.

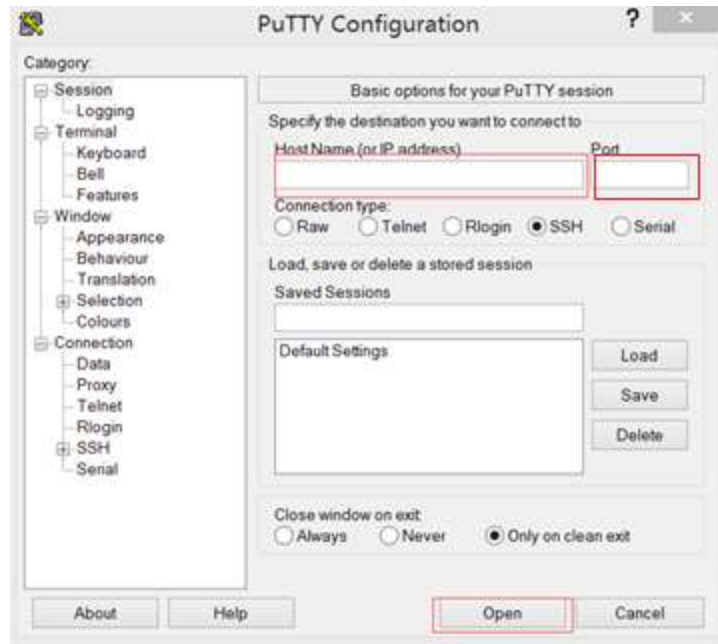
[Login server]

1. PuTTY is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator.

Download putty.exe at the following url:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

a) execute putty.exe Enter the IP address and click open



We are going to use the server in our department in this course.

The server IP is 10.20.13.130

The port is 10022

b) Enter a username and password

You should use the username which is assigned for you.

If this is the first time you login to the server, your default password is 123456.

You can change it to your own password after you login. For this, you can make use of linux command “passwd”.

After you login, you can use “pwd” to check your current folder. Your home directory will be at /data/cs310/{your_login_name} and that will be your working folder.

Please do not create folder at other places.

[Shell script example]

common commands:

you can always look at the command information with the `--help` option,

e.g. `ls --help`

`mkdir --help`

`ls` : displays the contents of the current directory

`mkdir xxx`: create a folder named xxx

`touch xxx`: create a file named xxx

`cat xxx`: view file contents

`cd /path`: change to the path directory

`rm (-rf) file(folder)` : delete a file or folder

`vi/vim`: use the vim editor to edit the file

`chmod`: modify file permissions

such as : `chmod +x file` (make files executable)

Variables:

Just about every programming language in existence has the concept of variables - a symbolic name for a chunk of memory to which we can assign values, read and manipulate its contents.

```
(base) fengpeng@nb1716:~$ MY_MESSAGE="Hello World"
(base) fengpeng@nb1716:~$ echo $MY_MESSAGE
Hello World
```

The above script assigns the string "Hello World" to the variable `MY_MESSAGE` and then echoes out the value of the variable. Note that we need the quotes around the string Hello World. Whereas we could get away with `echo Hello World` because `echo` will take any number of parameters, a variable can only hold one value, so a string with spaces must be quoted so that the shell knows to treat it all as one.

The shell does not care about types of variables; they may store strings, integers, real numbers - anything you like.

```
MY_MESSAGE="Hello World"
```

```
MY_SHORT_MESSAGE=hi
```

```
MY_NUMBER=1
```

```
MY_PI=3.142
```

```
MY_OTHER_PI="3.142"
```

```
MY_MIXED=123abc
```

We can interactively set variable names using the `read` command; the following script asks you for your name then greets you personally:

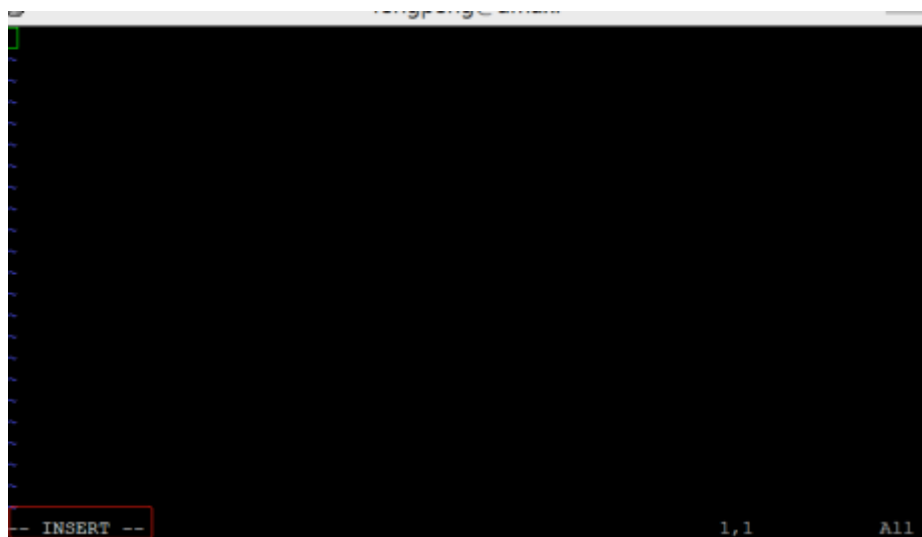
```
(base) fengpeng@nb1716:~$ read MY_NAME
ada
(base) fengpeng@nb1716:~$ echo "hello $MY_NAME"
hello ada
```

1) Create and edit a shell script

touch first.sh ; vi first.sh or
vi first.sh directly



Press 'i' on the keyword and then you can insert character in the shell script



2) write the first shell script

←:move cursor left
→:move cursor right
↓:move cursor down
↑:move cursor up

```
$ cat first.sh      view file contents

#!/bin/bash        Specify a script interpreter
echo "This is the first shell script"      output

echo "What is your name?"

read name          Keyboard input variable name

echo "Your name is $name"

if [ $name = "lilei" ]; then      if conditional statements
    echo "hello,lilei"
else
    echo "hello,$name"
fi
```

3) Save and exit

press [esc]

<pre>#!/bin/bash echo "This is the first shell script" echo "What is your name?" read name echo "My name is \$name" if [\$name = "lilei"]; then echo "hello,lilei" else echo "hello,\$name" fi ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ -- INSERT --</pre>	<pre>#!/bin/bash echo "This is the first shell script" echo "What is your name?" read name echo "My name is \$name" if [\$name = "lilei"]; then echo "hello,lilei" else echo "hello,\$name" fi ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~</pre>
---	--

You need to pay attention to the difference in the red box, the document changes from insert mode to normal mode when you press [esc]

Press :wq and enter key

```
#!/bin/bash
echo "This is the first shell script"

echo "What is your name?"

read name

echo "My name is $name"

if [ $name = "lilei" ]; then
    echo "hello,lilei"
else
    echo "hello,$name"
fi

~
~
~
~
~
~
~
~
~
~
:wq
```

4) execute shell script

Need to add executable permission and then run ./first.sh

```
(base) [REDACTED]@amax2:~$ chmod +x first.sh
(base) [REDACTED]@amax2:~$ ./first.sh
This is the first shell script
What is your name?
lilei
My name is lilei
hello,lilei
```

Exercise 1

Requirements:

- 1) Create a directory xxx in the current directory
- 2) Create a file named lab1.txt in xxx directory
- 3) Write "hello world" in this file
- 4) Write a shell script named lab1.sh in the current directory
- 5) Enter string "yes" or "no" when you run lab1.sh

6) Use conditional statements to judge, if the input is "yes", show the content of lab1.txt, otherwise output "nothing to do"

Exercise 2

In audio processing, we often need to give each utterance a distinct utterance id.

The utterance id is usually composed of a speaker id and a recording id. For example, an utterance id SUSTECH007_0017 means this utterance is spoken by speaker 007 and it is his 17th recording. SUSTECH is the brand name of the dataset.

If we have X speakers and each of them makes Y recordings, please write a shell script to generate all the utterance ids in ascending order. All the utterance ids need to have the same width. When X=30, Y=200, the output should look like:

```
SUSTECH001_0001
SUSTECH001_0002
...
SUSTECH001_0200
SUSTECH002_0001
...
SUSTECH030_0199
SUSTECH030_0200
```

The shell script should take two input arguments.

The usage should look like

```
./generate_uttid.sh X Y
```

,where X and Y are both integers.