

CSE 464 Software Quality Assurance and Testing

Lab - Introduction to UNIX Command Line and Java Program Compilation/Execution

Xusheng Xiao

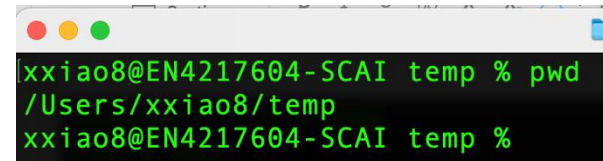
Associate Professor
School of Computing and Augmented Intelligence
Arizona State University

Bash in Terminal

- For Linux and Mac, simply open Terminal app
- For Windows
 - You may install Windows Subsystem for Linux (WSL)
 - <https://learn.microsoft.com/en-us/windows/wsl/install>
 - You may install VirtualBox and then use a Ubuntu virtual machine
 - <https://www.virtualbox.org/>
 - <https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview>

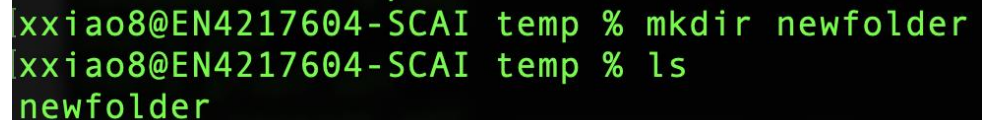
Working with Directories

- View your current directory
 - pwd (print working directory)



```
xxiao8@EN4217604-SCAI temp % pwd
/Users/xxiao8/temp
xxiao8@EN4217604-SCAI temp %
```

- Create directory
 - mkdir <directory>



```
xxiao8@EN4217604-SCAI temp % mkdir newfolder
xxiao8@EN4217604-SCAI temp % ls
newfolder
```

- Remove file/directory
 - rm <file>
 - rmdir <directory> - **directory needs to be empty**
 - rm -r <directory> - **all files and subdirectories will be removed!!**

Working with Directories – Cont.

- Directory path
 - Absolute path: /home/user – starting with /
 - Relative path: user – equivalent to /home/user if current directory is /home
- Change directory
 - `cd <absolute path>` or `cd <relative path>`
 - `cd ..` - go to parent directory
 - `cd ~` - go to the home directory

Create a file

- touch test

- ls -l

```
(base) xxiao8@en4217604-scai test % touch test
(base) xxiao8@en4217604-scai test % ls
file1  file2  test
(base) xxiao8@en4217604-scai test % ls -l
total 16
-rw-r--r--  1 xxiao8  staff   5 Sep  7 10:31 file1
-rw-r--r--  1 xxiao8  staff   6 Sep  7 10:31 file2
-rw-r--r--  1 xxiao8  staff   0 Sep  7 10:36 test
```

- vi test
- echo "helloworld" > test (redirection to file)

UNIX Access Control

- `ls -l`

- UNIX: 3 classes of users: owner, group, rest

- rwx rwx rwx

rest

group

owner

```
(base) xxiao8@en4217604-scai test % ls -l
total 16
-rw-r--r--  1 xxiao8  staff  5 Sep  7 10:31 file1
-rw-r--r--  1 xxiao8  staff  6 Sep  7 10:31 file2
(base) xxiao8@en4217604-scai test %
```

of links

user

group

size (bytes)

modification
date and time

Grant Access Controls using Symbolic Mode

- `chmod u+x testFile`
 - *u* stands for user.
 - *g* stands for group.
 - *o* stands for others.
 - *a* stands for all.
 - *r*, *w*, *x* stand for read, write, execute
- `echo "echo 'Hello World' " > testFile`
- `./testFile` (execute the script)

Grant Access Controls via Three-Digit Code

- Three-digit code for permissions:
 - 0 = no permissions whatsoever; this person cannot read, write, or execute the file
 - 1 = execute only
 - 2 = write only
 - 3 = write and execute (1+2)
 - 4 = read only
 - 5 = read and execute (4+1)
 - 6 = read and write (4+2)
 - 7 = read and write and execute (4+2+1)
- Examples
 - `chmod 700 test` -> only owner can read, write, execute
 - `chmod 777 test` -> every one can read, write, execute
 - `chmod 744 test` -> only owner can read, write, execute; every one else can read the file only

Copy and Move Files

- Copy a file
 - cp source destination

```
[xxiao8@EN4217604-SCAI test2 % cp data data2  
[xxiao8@EN4217604-SCAI test2 % ls -l  
total 16  
-rw-r--r--  1 xxiao8  staff  5 Aug 27 10:09 data  
-rw-r--r--  1 xxiao8  staff  5 Aug 27 10:09 data2
```

- Move a file
 - mv source destination (i.e., cut and paste)

```
[xxiao8@EN4217604-SCAI test2 % mv data subfolder  
[xxiao8@EN4217604-SCAI test2 % cd subfolder  
[xxiao8@EN4217604-SCAI subfolder % ls -l  
total 8  
-rw-r--r--  1 xxiao8  staff  5 Aug 27 10:09 data
```

Java Environment

- To run Java, you need to install JRE (Java Runtime)
 - <https://www.java.com/en/download/manual.jsp>
- To compile Java programs, you need to install JDK
 - <https://www.oracle.com/java/technologies/downloads/>
 - <https://docs.oracle.com/en/java/javase/22/install/overview-jdk-installation.html>
 - Test your installation: type 'javac' in your terminal

```
xxiao8@EN4217604-SCAI temp % javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                Read options and filenames from file
  -Akey[=value]              Options to pass to annotation processors
  --add-modules <module>(<module>)*
                             Root modules to resolve in addition to the initial modules, or all modules
                             on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                             Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                             Specify where to find user class files and annotation processors
  -d <directory>             Specify where to place generated class files
```

First Java Program – “Hello World”

- Create a file named HelloWorld.java:
- Edit the file to print out “Hello World!”:

```
class HelloWorld {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello World!");  
    }  
}
```

- Save the file.

Compile and Run Java Program

- Type “javac HelloWorld.java”

```
xxiao8@EN4217604-SCAI temp % javac HelloWorld.java
```

- Check the compiled file “HelloWorld.class”

```
xxiao8@EN4217604-SCAI temp % ls -l
total 16
-rw-r--r--@ 1 xxiao8  staff  426 Aug 26 09:47 HelloWorld.class
-rw-r--r--  1 xxiao8  staff  103 Aug 26 09:47 HelloWorld.java
```

- Type “java HelloWorld” (**not HelloWorld.class**) to run the compiled program:

```
xxiao8@EN4217604-SCAI temp % java HelloWorld
Hello World!
```

Create Package for Java Program

- Move the HelloWorld.java to a folder called lib
- Add “package lib;” at the beginning of the file

```
package lib;  
class HelloWorld {  
  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello World!");  
    }  
  
}
```

- Compile using “javac lib/HelloWorld.java”

Create Jar File

- Jar stands for Java ARchive.
- It's a file format based on the popular ZIP file format and is used for aggregating many files into one.
- Type “jar cfe HelloWorld.jar lib.HelloWorld lib/HelloWorld.class” to create a jar file with the main class being HelloWorld
 - c option: create a jar file
 - f option: specify a jar file name
 - e option: specify a main class

```
xxiao8@EN4217604-SCAI temp % jar cfe HelloWorld.jar lib.HelloWorld lib/HelloWorld.class
xxiao8@EN4217604-SCAI temp % ls -l
total 40
-rw-r--r--@ 1 xxiao8  staff  789 Aug 26 22:30 HelloWorld.jar
```

Use Jar File

- Run Jar file that contains a main class
 - Type “java -jar HelloWorld.jar”

```
xxiao8@EN4217604-SCAI temp % java -jar HelloWorld.jar  
Hello World!
```

- Use a Jar file as a library
 - Create a new Java program HelloWorldUse in another directory test
 - We would like to use the HelloWorld program we just created
 - Create a HelloWorld class instance and invoke its main function
 - Note that we need “import lib.HelloWorld;” at the beginning

```
import lib.HelloWorld;  
public class HelloWorldUse {  
    public static void main(String[] args) throws Exception{  
        new HelloWorld().main(new String[0]);  
    }  
}
```

Use Jar File – Cont.

- Copy the HelloWorld.jar to the test folder
- Compile using “javac -cp HelloWorld.jar HelloWorldUse.java”
 - Use the classpath option -cp to specify the jar file for compilation

```
xxiao8@EN4217604-SCAI test % javac -cp HelloWorld.jar HelloWorldUse.java
xxiao8@EN4217604-SCAI test % ls -l
total 24
-rw-r--r--@ 1 xxiao8  staff   788 Aug 26 22:41 HelloWorld.jar
-rw-r--r--@ 1 xxiao8  staff   347 Aug 26 22:42 HelloWorldUse.class
-rw-r--r--  1 xxiao8  staff   141 Aug 26 22:40 HelloWorldUse.java
```

- Run “java -cp “HelloWorld.jar:.” HelloWorldUse”.
 - Use the classpath option -cp to specify the jar file and the current folder “.” for running the Java program

```
xxiao8@EN4217604-SCAI test % java -cp "HelloWorld.jar:." HelloWorldUse
Hello World!
```


Thank You !



Questions ?