Name: Timothy Caole
Date: 1/20/2024

# Program1J

## Shell.java Report

1) **Explain your design and the algorithm for your Shell.java, support your explanation using flowcharts or other figures.**

The file Shell.java is suppose to mimic what a shell program does in an operating system, serving as a command-line interface, allowing users to interact with the operating system by entering textual commands to execute various.

Figure 1.1 – Shell.java Design

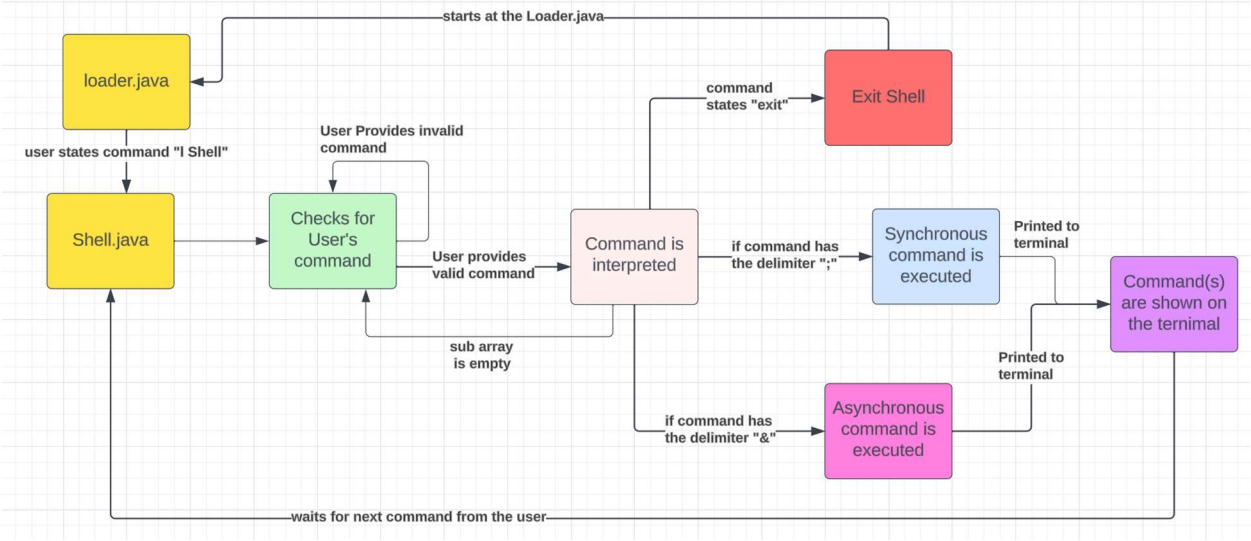



Figure 1.2 –ThreadOS File interaction

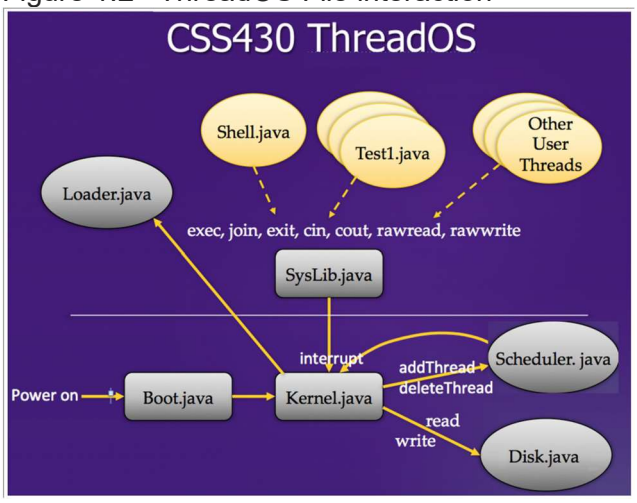
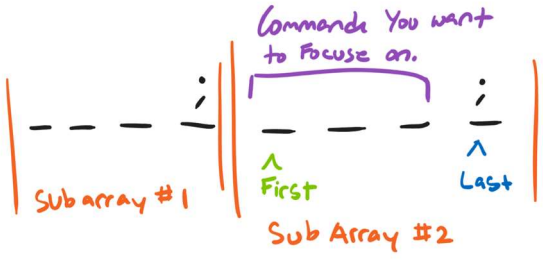



Figure 1.3 – generateCmd() Method Visuals



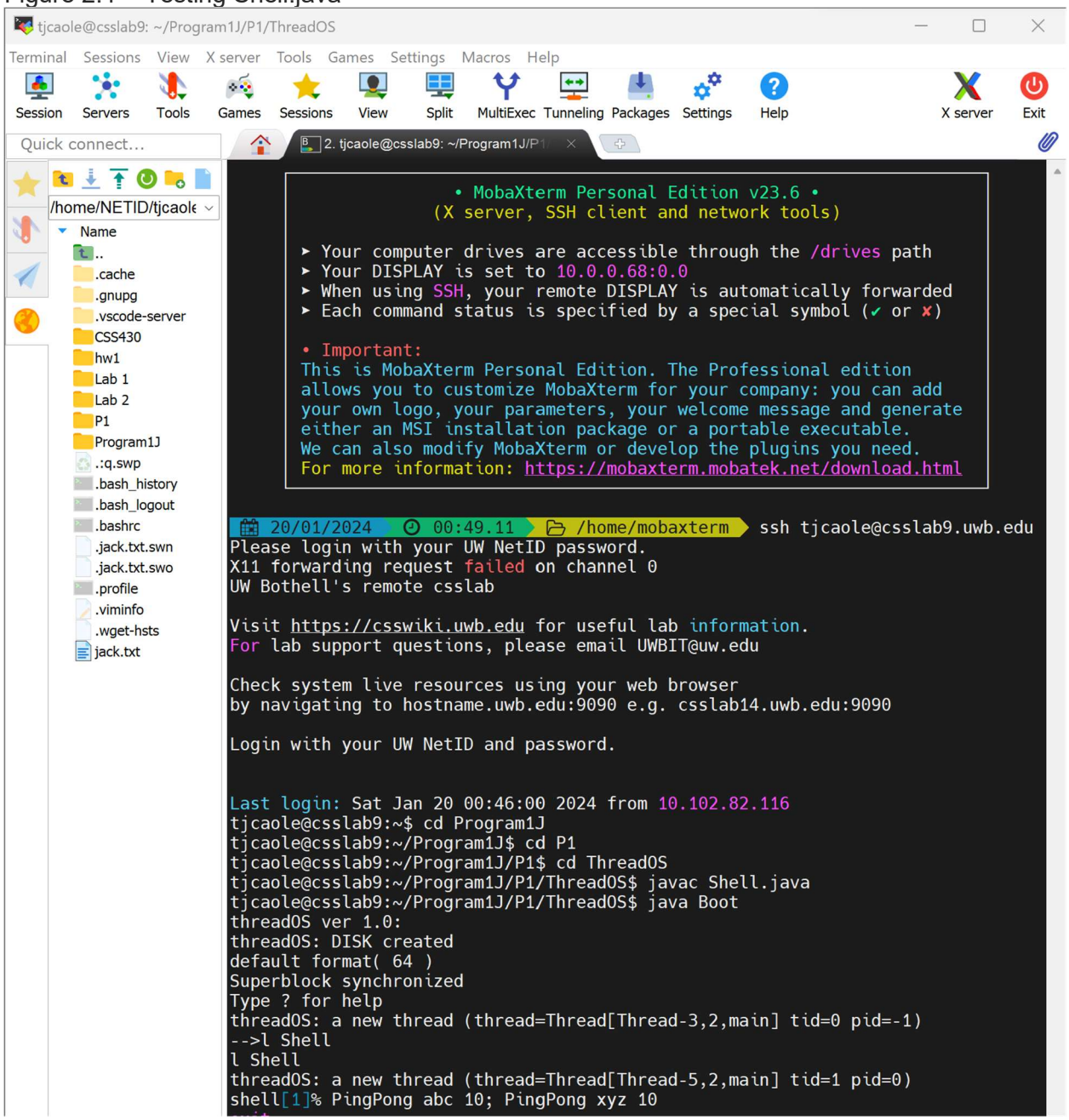

Seen in Figure 1.1, is the Shell.java file design, the Shell.java file will take a command from the user that has typed a command in a terminal. That command will be inputted as a buffer then converted to a string and stored in an array of strings. SysLib functions from the SysLib.java file will be used to interact with the Kernal, seen in figure 1.2, and execute the commands. The commands are split into 3 section, exit, synchronous, or asynchronous. Synchronous

# Program1J

commands are determined if it has a ";" delimiter, and an asynchronous commands are determined if it has a "&" delimiter. When the command is exit, the program will terminal.

**2) Explain how to test your Shell.java.**

Figure 2.1 – Testing Shell.java




Seen in Figure 2.1, to test Shell.java, open MobaXterm, log into csslab using your ssh <netID>@csslab<9-12>.uwb. and your netid password.
Find the folder where you have Shell.java. In this case folder "ThreadOS". Compile the file to make sure there isn't any errors by typing "javac Shell.java". The Type "java Boot","l Shell ", and then type your command.

Name: Timothy Caole
Date: 1/20/2024

# Program1J

**3. Output: Include *screenshots* of the output from testing your Shell.java as stated above in the assignment description**

Figure 3.1 – Exit

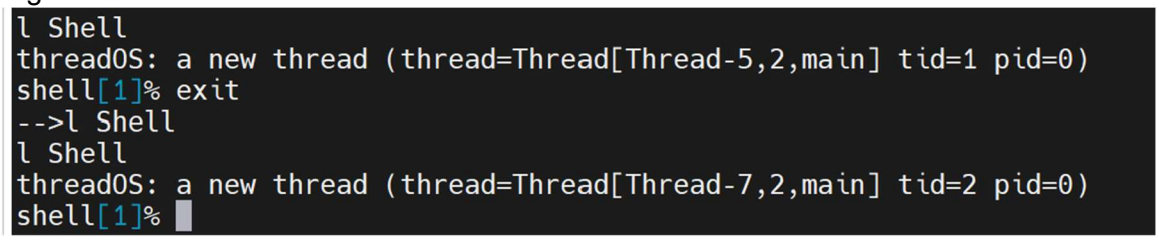



Figure 3.2 – Single Synchronous Commands

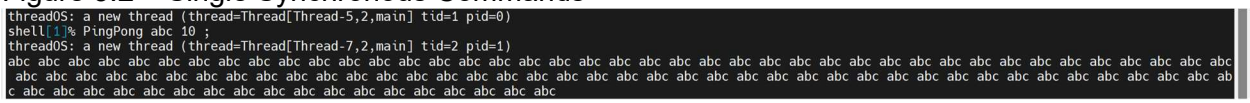



Figure 3.3 –Single Asynchronous Commands

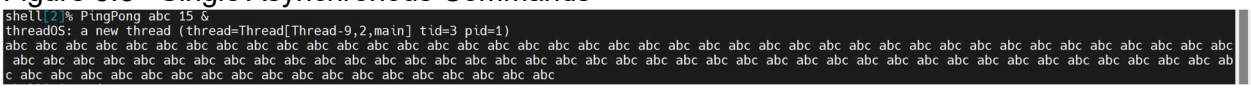
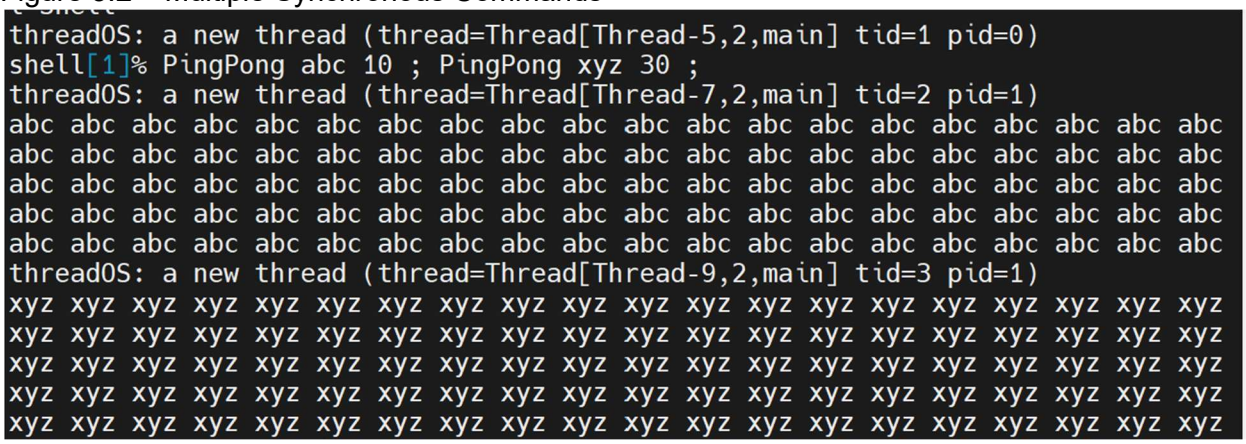



Figure 3.2 – Multiple Synchronous Commands



Name: Timothy Caole
Date: 1/20/2024

# Program1J

Figure 3.3 – Multiple Asynchronous Commands

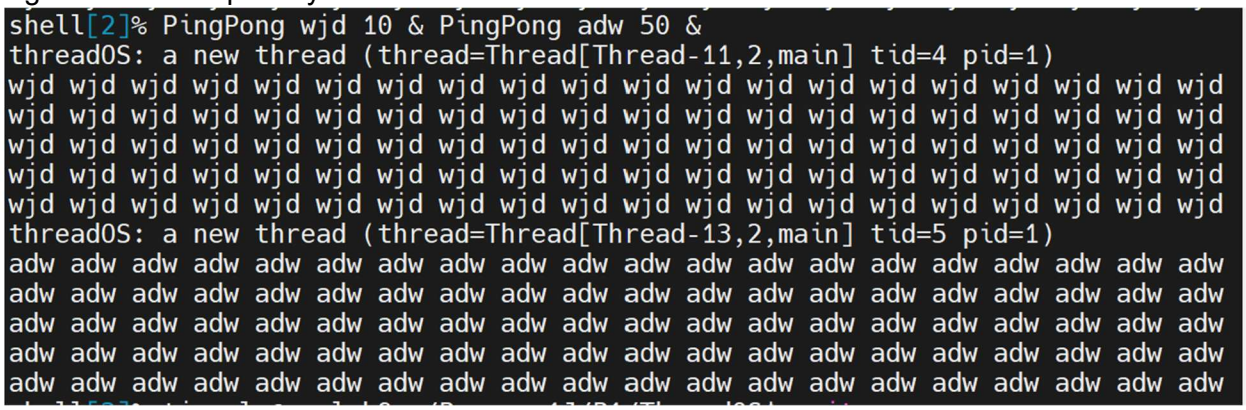
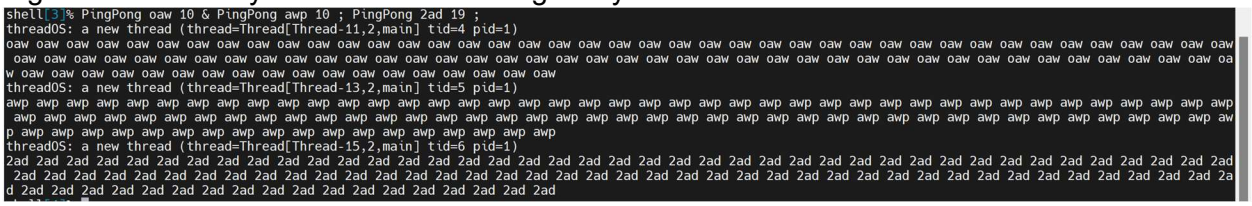



Figure 3.4 – Two Synchronous and a Single Asynchronous Commands



Name: Timothy Caole
Date: 1/20/2024

# Program1J

## Notes:

1) Open MobaXterm, don't forget to connect Big-IP Edge Client
2) Log in using the following,:

ssh <netID>@csslab<9-12>.uwb.edu E.g: ssh tjcaole@csslab9.uwb.edu
password: your netid password E.g

3) To extract the files for our homework use the following command, note command is case sensitive.

"Cp -r /usr/apps/CSS430/ThreadOS /home/NETID/YOURNETIDHERE/FOLDERNAMEHERE"
E.g: Cp -r /usr/apps/CSS430/ThreadOS /home/NETID/tjcaole /P1

4) The folders that was copied, drag and drop it into your ide (in this case intellij)

5) There will be an error that SysLib won't work, to fix that look at the solution below.

*Error:*
** Using intellij IDE*
** [solved] Error w/ SysLib: cannot resolve symbol*
** Solution:*
*https://washington.zoom.us/rec/play/8jw0cAwQInMUM3V6sAmWRQJ_cSrfr5fqKN3OsEsa5N1MXG0_SFA8i1eJCPEwpVkYFdaXR4-W8NONcsBm.W4FjTN7pMo6cB-_d*
** select File > Program Structure > Module > Dependencies > + > Path of project AKA P1> JARs or Directories >*
** add the path to the ThreadOS folder then click apply and ok.*

6) Move the Src files into the ThreadOS folder to make things easier.

7) When you want to compile your shell.java. drag and drop the entire folder (P1) into MobaXterm

Type the following command in mobaxterm "javac Shell.java", cap sensitive, in the folder it is stored it.

8) To run the program

Type the following command in mobaxterm "java Boot", and "I Shell " and then type a command.
Command E.g: "PingPong abc 10; PingPong xyz 10"