

## 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

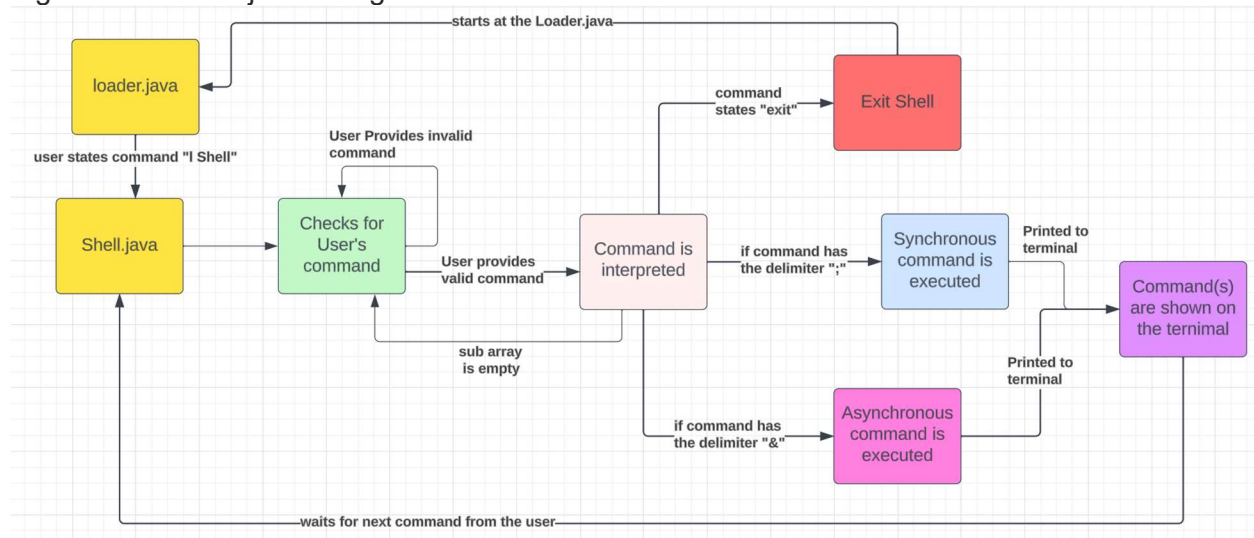


Figure 1.2 – ThreadOS File interaction

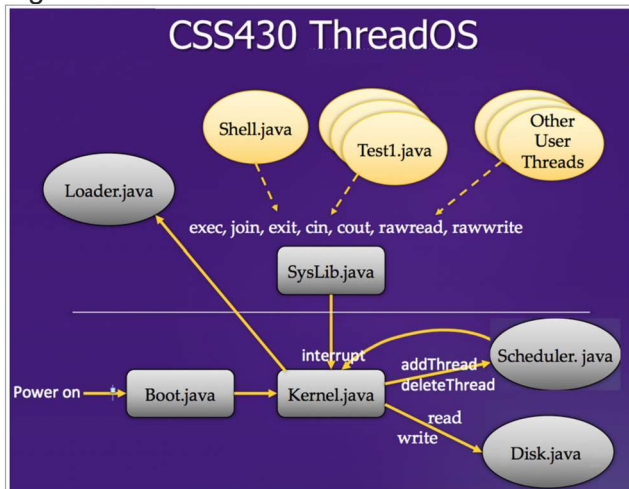
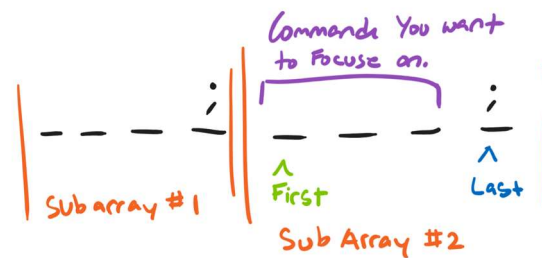


Figure 1.3 – generateCmd() Method Visuals



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

Name: Timothy Caole

Date: 1/20/2024

## 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

The screenshot shows a MobaXterm window titled "tjcaole@csslab9: ~/Program1J/P1/ThreadOS". The left sidebar displays a file explorer view of the local system, showing the path /home/NETID/tjcaole and various files and folders. The main terminal area displays the MobaXterm welcome message, which includes information about the version (v23.6) and features like X server, SSH client, and network tools. It also provides important instructions for using the software, such as setting the DISPLAY environment variable and forwarding SSH connections. The terminal output shows the user logging in via SSH, navigating to the Program1J directory, and running the javac command to compile Shell.java. The compilation is successful, and the user then runs the java Boot command, which starts the ThreadOS application. The application output shows the creation of a disk, synchronization of the superblock, and the start of a new thread. The user then enters the command "PingPong abc 10; PingPong xyz 10", which is executed successfully.

```
tjcaole@csslab9: ~/Program1J/P1/ThreadOS

• MobaXterm Personal Edition v23.6 •
(X server, SSH client and network tools)

> Your computer drives are accessible through the /drives path
> Your DISPLAY is set to 10.0.0.68:0.0
> When using SSH, your remote DISPLAY is automatically forwarded
> Each command status is specified by a special symbol (✓ or ✗)

• Important:
This is MobaXterm Personal Edition. The Professional edition
allows you to customize MobaXterm for your company: you can add
your own logo, your parameters, your welcome message and generate
either an MSI installation package or a portable executable.
We can also modify MobaXterm or develop the plugins you need.
For more information: https://mobaxterm.mobatek.net/download.html

20/01/2024 00:49.11 /home/mobaxterm ssh tjcaole@csslab9.uwb.edu
Please login with your UW NetID password.
X11 forwarding request failed on channel 0
UW Bothell's remote csslab

Visit https://csswiki.uwb.edu for useful lab information.
For lab support questions, please email UWBIT@uw.edu

Check system live resources using your web browser
by navigating to hostname.uwb.edu:9090 e.g. csslab14.uwb.edu:9090

Login with your UW NetID and password.

Last login: Sat Jan 20 00:46:00 2024 from 10.102.82.116
tjcaole@csslab9:~$ cd Program1J
tjcaole@csslab9:~/Program1J$ cd P1
tjcaole@csslab9:~/Program1J/P1$ cd ThreadOS
tjcaole@csslab9:~/Program1J/P1/ThreadOS$ javac Shell.java
tjcaole@csslab9:~/Program1J/P1/ThreadOS$ java Boot
threadOS ver 1.0:
threadOS: DISK created
default format( 64 )
Superblock synchronized
Type ? for help
threadOS: a new thread (thread=Thread[Thread-3,2,main] tid=0 pid=-1)
-->l Shell
l Shell
threadOS: a new thread (thread=Thread[Thread-5,2,main] tid=1 pid=0)
shell[1]% PingPong abc 10; PingPong xyz 10
```

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.

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

```
l Shell
thread0S: a new thread (thread=Thread[Thread-5,2,main] tid=1 pid=0)
shell[1]% exit
-->l Shell
l Shell
thread0S: a new thread (thread=Thread[Thread-7,2,main] tid=2 pid=0)
shell[1]%
```

### Figure 3.2 – Single Synchronous Commands

[illegible]

### Figure 3.3 –Single Asynchronous Commands

[illegible]

### Figure 3.2 – Multiple Synchronous Commands

[illegible]





## 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/8jw0cAwQlnMUM3V6sAmWRQJ\\_cSrfr5fqKN3OsEsa5N1MXG0\\_SFA8i1eJCPEwpVkYFdaXR4-W8NONcsBm.W4FjTN7pMo6cB-\\_d](https://washington.zoom.us/rec/play/8jw0cAwQlnMUM3V6sAmWRQJ_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"