Name: Thuan Tran
Date: April 7th, 2017
Program 1

REPORT

Table of Contents

The task

This assignment is a brief overview of the ThreadOS system that the students will build incrementally over the course.
For this assignment there are two tasks that the students need to do:

- Implement processes.cpp by using pipe and dup2
- Implement Shell.java to act as a Shell Interpreter for ThreadOS

Processes.cpp

Processes.cpp simulate the way the command line interpret commands. First, it will create a processes of its own, then for that processes, it will create additional processes to carry out the commands.

Each processes is created by using fork() . It will return the ID of the processes. The method excelp() execute the command that is passed in. And dup2() duplicate or overide the file descriptor (cin or cout) for communication between processes.

For this assignment, we were asked to implement :

`“ps -A | grep argv[1] | wc -l”` where argv[1] is the passed argument from the command line

For these commands, the latest processes will be the one that execute the first command, then it will propagate up to the hierarchy where the first child process execute the last commands

Algorithm

For this commands, I decided to use 2 pipe where the first command communicate with the second command through the first page and the second command communicate to the third command through the second pipe.

From the main method, I first fork a process (Child) then I waited until the Child executed. Inside a child, I created another process (Grand Child). For the Child process, I also wait until the Grand Child finished. And in the Grand Child, I create another process (Great Grand Child) and make the Grand Child wait for the Great Grand Child. Inside Greate Grand Child, I execute the first command, pass the information through the pipe. After the Great Grand Child finished, The Grand Child received the information for the Great Grand Child, execute the second command and passed the information through another pipe. And inside Child where it received the info from Grand Child, It executed the last command and exit. At this point, the main method has finished waiting for the child then exit

During the ways, I also put additional check statements to see if the fork is created successfully or if the pipe is created

How I Test

I print out some information that I put into the child, grand child and great grand child to indicate where in the program the process is at

Shell.java

Shell.java simulates the shell of the ThreadOS system. It loads application and execute them in a sequential or concurrent way denoted by adding “&” or “;” between command.

Algorithm

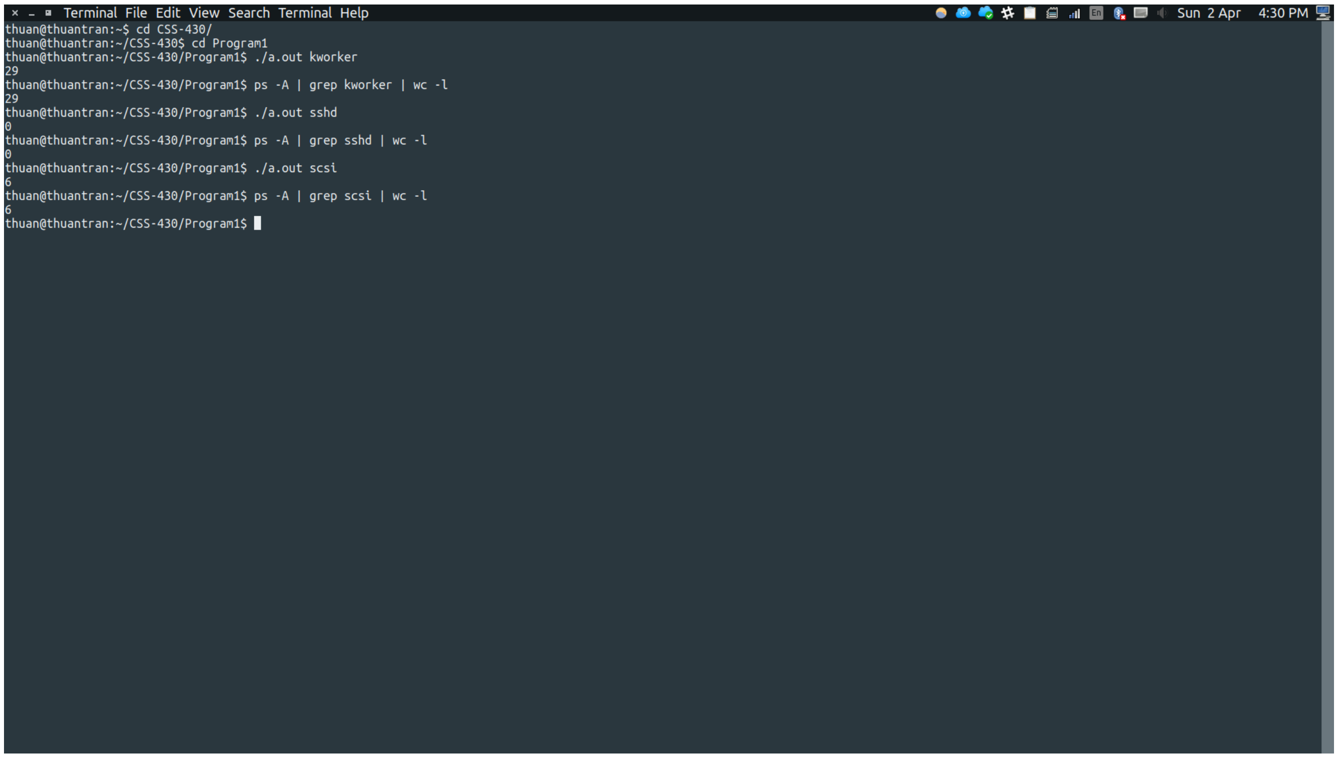
When the Shell.java is compiled and loaded, it will enter an infinite while loop until a user typed exit. Then algorithm will then get the input from the keyboard and seperate them by using the delimter “;” that serve as a sequential command. This means that the commands is now separated and will not have any interact with each other. Hence, we do not need to worry about concurrent

The separated command is now passed into another method. This method also separate the command by using the delimter “&” to check how many concurrent command are there. Then it will continue to execute each command. After they are done, the method loop through again to make sure all the concurrent command is executed before return the method to execute the next sequential commands

How I test

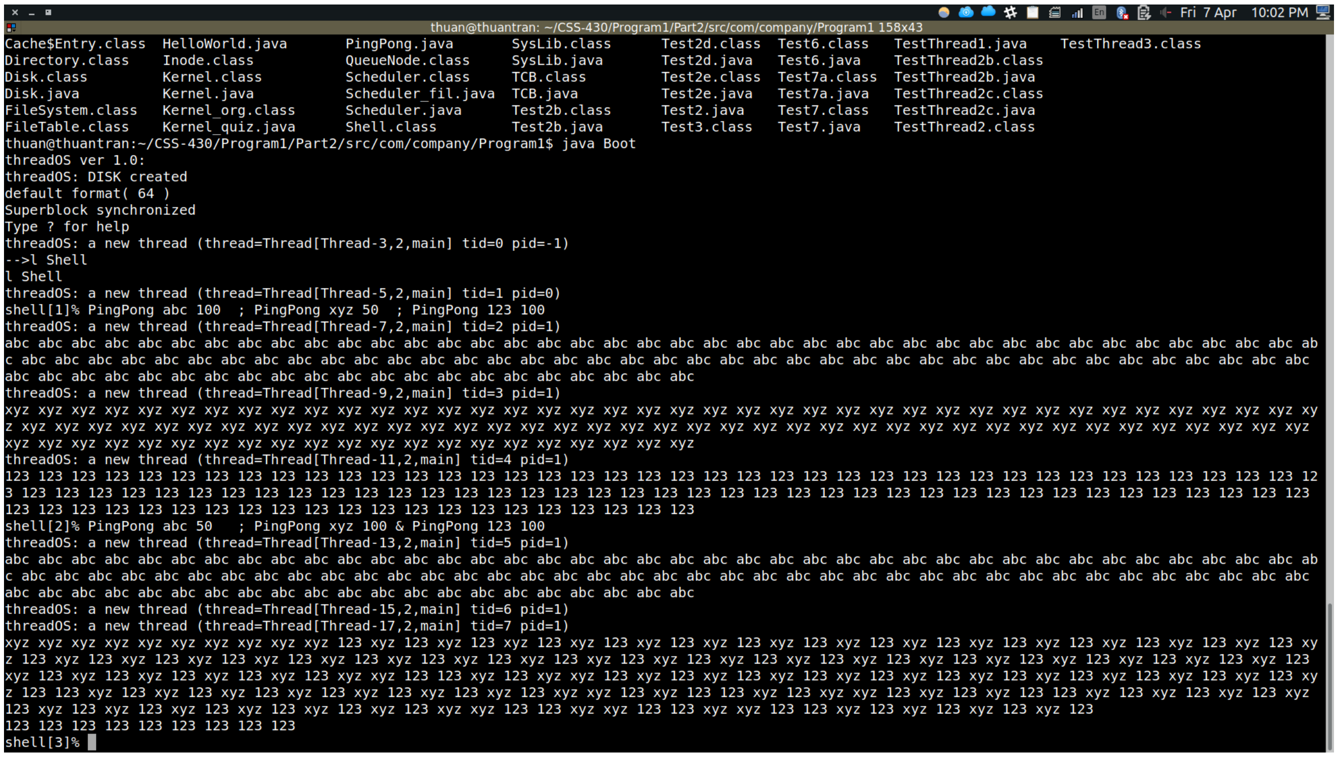
Just like above, I tested it by inserting multiple cout statements. I also created a regular expression to express the command and decided how to separate them.

Output



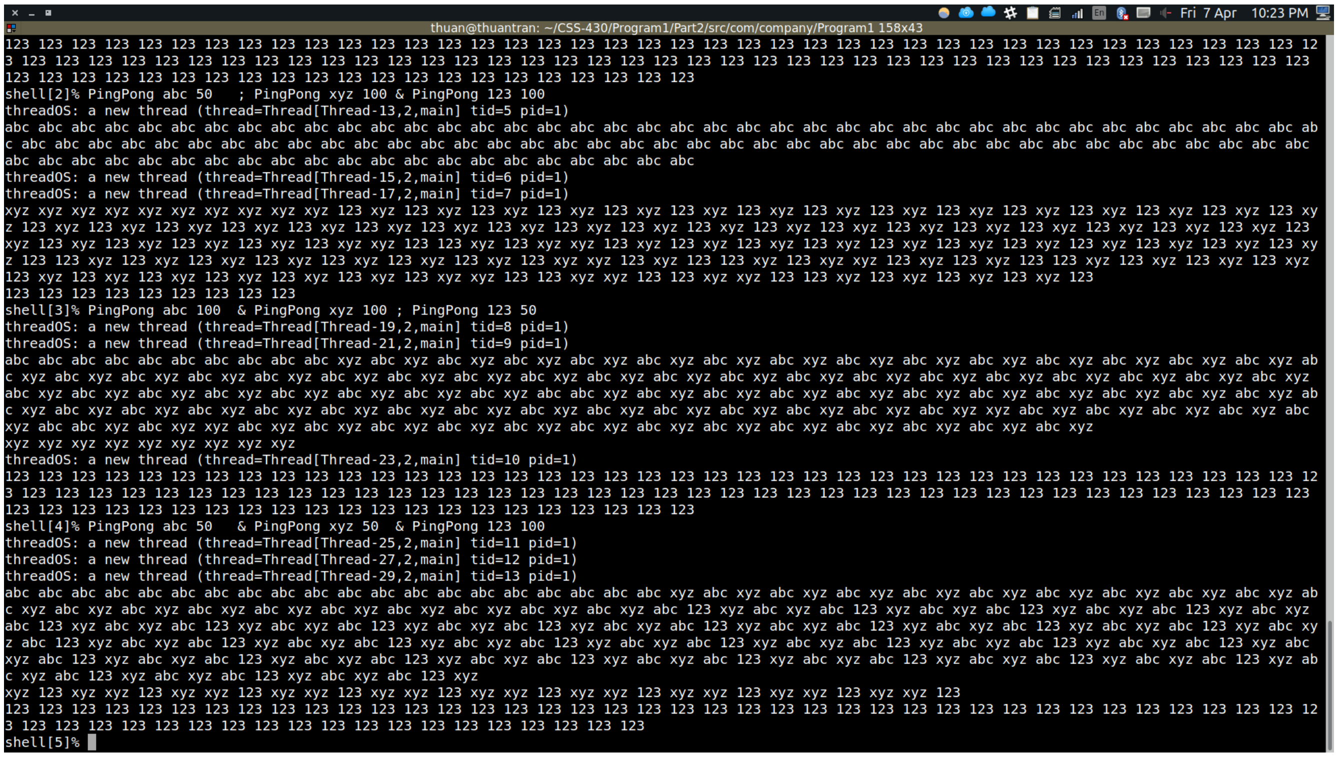
```
thuan@thuantran:~$ cd CSS-430/
thuan@thuantran:~/CSS-430$ cd Program1
thuan@thuantran:~/CSS-430/Program1$ ./a.out kworker
29
thuan@thuantran:~/CSS-430/Program1$ ps -A | grep kworker | wc -l
29
thuan@thuantran:~/CSS-430/Program1$ ./a.out sshd
0
thuan@thuantran:~/CSS-430/Program1$ ps -A | grep sshd | wc -l
0
thuan@thuantran:~/CSS-430/Program1$ ./a.out scsi
6
thuan@thuantran:~/CSS-430/Program1$ ps -A | grep scsi | wc -l
6
thuan@thuantran:~/CSS-430/Program1$ 
```

Illustration 1: Output of Process.cpp compared to Command Line



```
Cache$Entry.class  HelloWorld.java       PingPong.java       SysLib.class     Test2d.class  Test6.class   TestThread1.java    TestThread3.class
Directory.class    Inode.class           QueueNode.class     SysLib.java      Test2d.java   Test6.java    TestThread2b.class
Disk.class         Kernel.class          Scheduler.class     TCB.class        Test2e.class  Test7a.class  TestThread2b.java
Disk.java          Kernel.java           Scheduler_fil.java  TCB.java         Test2e.java   Test7a.java   TestThread2c.class
FileSystem.class   Kernel_org.class      Scheduler.java      Test2b.class     Test2.java    Test7.class   TestThread2c.java
FileTable.class    Kernel_quiz.java      Shell.class         Test2b.java      Test3.class   Test7.java    TestThread2.class
thuan@thuantran:~/CSS-430/Program1/Part2/src/com/company/Program1$ 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 100  ; PingPong xyz 50  ; PingPong 123 100
threadOS: a new thread (thread=Thread[Thread-7,2,main] tid=2 pid=1)
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc ab
c abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
threadOS: a new thread (thread=Thread[Thread-9,2,main] tid=3 pid=1)
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xy
z xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz
threadOS: a new thread (thread=Thread[Thread-11,2,main] tid=4 pid=1)
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 12
3 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
shell[2]% PingPong abc 50   ; PingPong xyz 100 & PingPong 123 100
threadOS: a new thread (thread=Thread[Thread-13,2,main] tid=5 pid=1)
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc ab
c abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
threadOS: a new thread (thread=Thread[Thread-15,2,main] tid=6 pid=1)
threadOS: a new thread (thread=Thread[Thread-17,2,main] tid=7 pid=1)
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xy
z 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123
xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz xyz 123 123 xyz 123 xyz xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xy
z 123 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz xyz 123 xyz 123 123 xyz 123 xyz xyz 123 xyz 123 xyz 123 123 xyz 123 xyz 123 xyz 123 xyz
123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz xyz 123 123 xyz xyz 123 123 xyz xyz 123 123 xyz 123 xyz 123 xyz 123 xyz 123
123 123 123 123 123 123 123 123 123
shell[3]% 
```

Illustration 2: Output of Shell.java running the first 2 test cases


Fri 7 Apr 10:23 PM
thuan@thuantran: ~/CSS-430/Program1/Part2/src/com/company/Program1 158x43

```
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 12
3 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
shell[2]% PingPong abc 50   ; PingPong xyz 100 & PingPong 123 100
threadOS: a new thread (thread=Thread[Thread-13,2,main] tid=5 pid=1)
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc ab
c abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
threadOS: a new thread (thread=Thread[Thread-15,2,main] tid=6 pid=1)
threadOS: a new thread (thread=Thread[Thread-17,2,main] tid=7 pid=1)
xyz xyz xyz xyz xyz xyz xyz xyz xyz xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xy
z 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123
xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz xyz 123 123 xyz 123 xyz xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xy
z 123 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz xyz 123 xyz 123 123 xyz 123 xyz xyz 123 xyz 123 xyz 123 123 xyz 123 xyz 123 xyz 123 xyz
123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz 123 xyz xyz 123 123 xyz xyz 123 123 xyz xyz 123 123 xyz 123 xyz 123 xyz 123 xyz 123
123 123 123 123 123 123 123 123 123
shell[3]% PingPong abc 100  & PingPong xyz 100 ; PingPong 123 50
threadOS: a new thread (thread=Thread[Thread-19,2,main] tid=8 pid=1)
threadOS: a new thread (thread=Thread[Thread-21,2,main] tid=9 pid=1)
abc abc abc abc abc abc abc abc abc abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz ab
c xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz
abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz ab
c xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz xyz abc xyz abc xyz abc xyz abc
xyz abc abc xyz abc xyz xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz
xyz xyz xyz xyz xyz xyz xyz xyz xyz
threadOS: a new thread (thread=Thread[Thread-23,2,main] tid=10 pid=1)
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 12
3 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
shell[4]% PingPong abc 50   & PingPong xyz 50  & PingPong 123 100
threadOS: a new thread (thread=Thread[Thread-25,2,main] tid=11 pid=1)
threadOS: a new thread (thread=Thread[Thread-27,2,main] tid=12 pid=1)
threadOS: a new thread (thread=Thread[Thread-29,2,main] tid=13 pid=1)
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz ab
c xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz
abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xy
z abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc
xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz ab
c xyz abc 123 xyz abc xyz abc 123 xyz abc xyz abc 123 xyz
xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123 xyz xyz 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 12
3 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
shell[5]% 
```

Illustration 3: Output of Shell.java with the last 2 test cases