Angie(Nomingerel) Tserenjav
CSS 430
Section A
13 October 2019

Program1

Table of Contents

Part 1: Linux System Programming

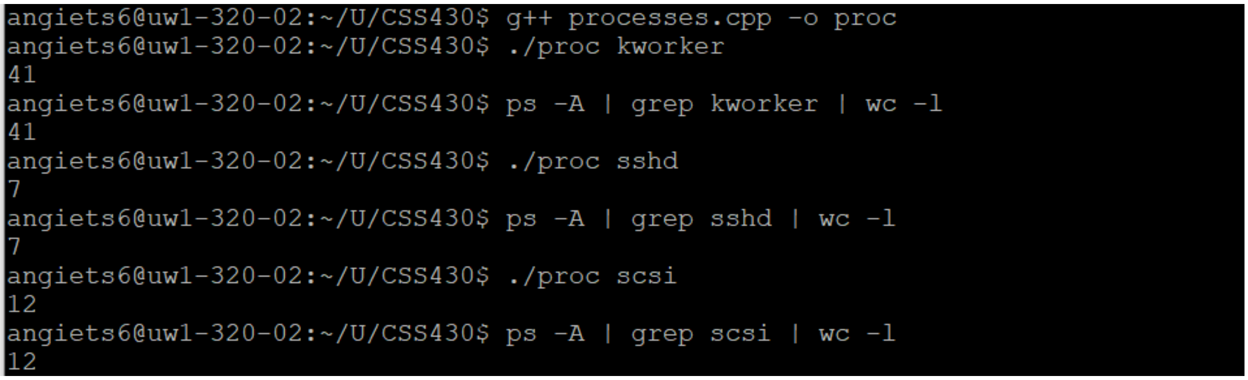
For this part, the processes.cpp program has been implemented and receives one argument. First, it creates two pipes which is pipefd1[2] and pipefd[2], the first pipe is connected between great grand child and grand child, and the other is to connect between grandchild and child. After that, it will create the new child process calling fork() system call. It will check the error cases such as if the pipefd1 is < 0 or if pipefd2 is < 0, then it will display a pipe failed error. It will create the parent process and move to child process, if it returns 0 it will move to the grand child process and then it will call the child process. In addition, dup2 is duplicate the file descriptor for communication between processes.

Following is the output of the processes.cpp running the programs:

- $ processes kworker
- $ ps -A | grep kworker | wc -l

- $ processes sshd
- $ ps -A | grep sshd | wc -l

- $ processes scsi
- $ ps -A | grep scsi | wc -l



```
angiets6@uw1-320-02:~/U/CSS430$ g++ processes.cpp -o proc
angiets6@uw1-320-02:~/U/CSS430$ ./proc kworker
41
angiets6@uw1-320-02:~/U/CSS430$ ps -A | grep kworker | wc -l
41
angiets6@uw1-320-02:~/U/CSS430$ ./proc sshd
7
angiets6@uw1-320-02:~/U/CSS430$ ps -A | grep sshd | wc -l
7
angiets6@uw1-320-02:~/U/CSS430$ ./proc scsi
12
angiets6@uw1-320-02:~/U/CSS430$ ps -A | grep scsi | wc -l
12
```

Part 2: ThreadOS Shell Design

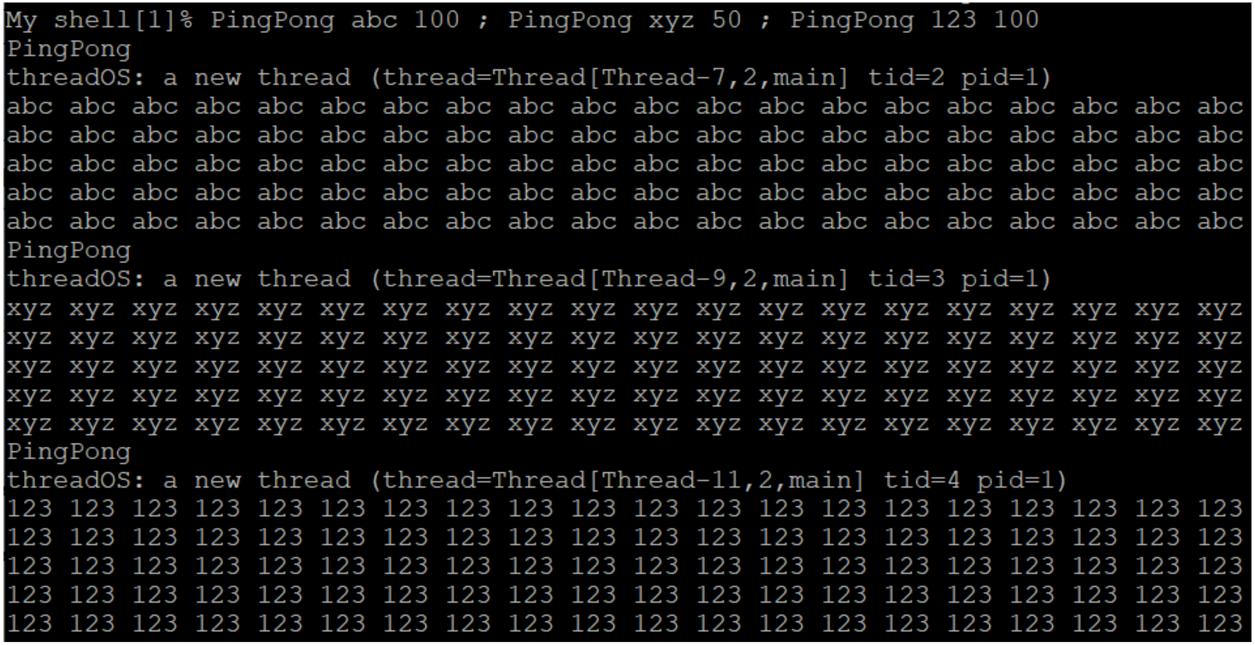
The Shell. java program has been implemented. This file allows to test the several PingPong shell files split with semicolon and ampersand. This program will execute public void run first. The run function lineCount is set to 1 so that it will represent the current line count. If the user input is only with split semicolon then it will call the command handler 2. In the handler2 function it will get ‘;’ argument and execute the program. If the user input is split with ‘&’, then the program will call handler 1 function. In the handler1 function, it takes the ‘&’ as the argument and executes the program. On the other hand, if the user input is both ‘;’ and ‘&’, then it will call the multi function. In the multi function, it will check if the there any splits other than ‘&’ then executes the ‘;’ function.

Step by step instructions to run Shell.java program:

1. Check if the Shell.java is in the same folder as ThreadOS.
2. Compile the Shell.java using javac Shell.java command
3. Run the java Boot
4. Execute -l Shell command
5. When My Shell[1]% displays: Test
   PingPong abc 100 ; PingPong xyz 50 ; PingPong 123 100

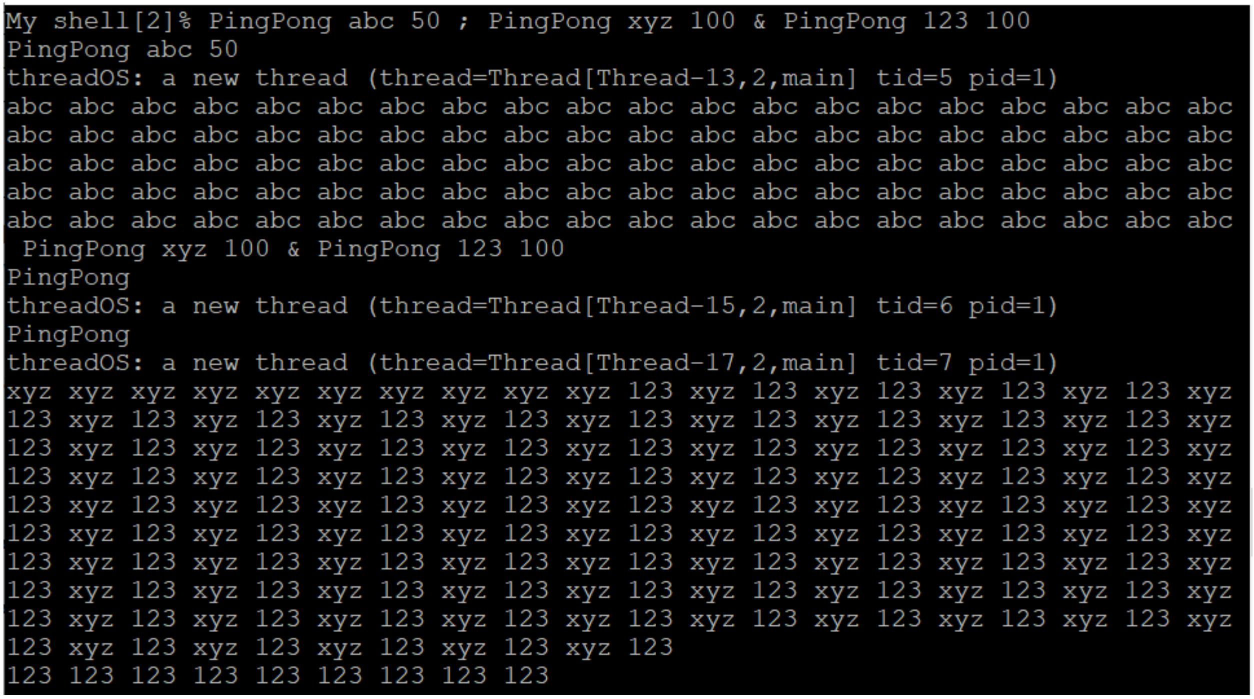
Test Output:

My Shell[1]% Output:



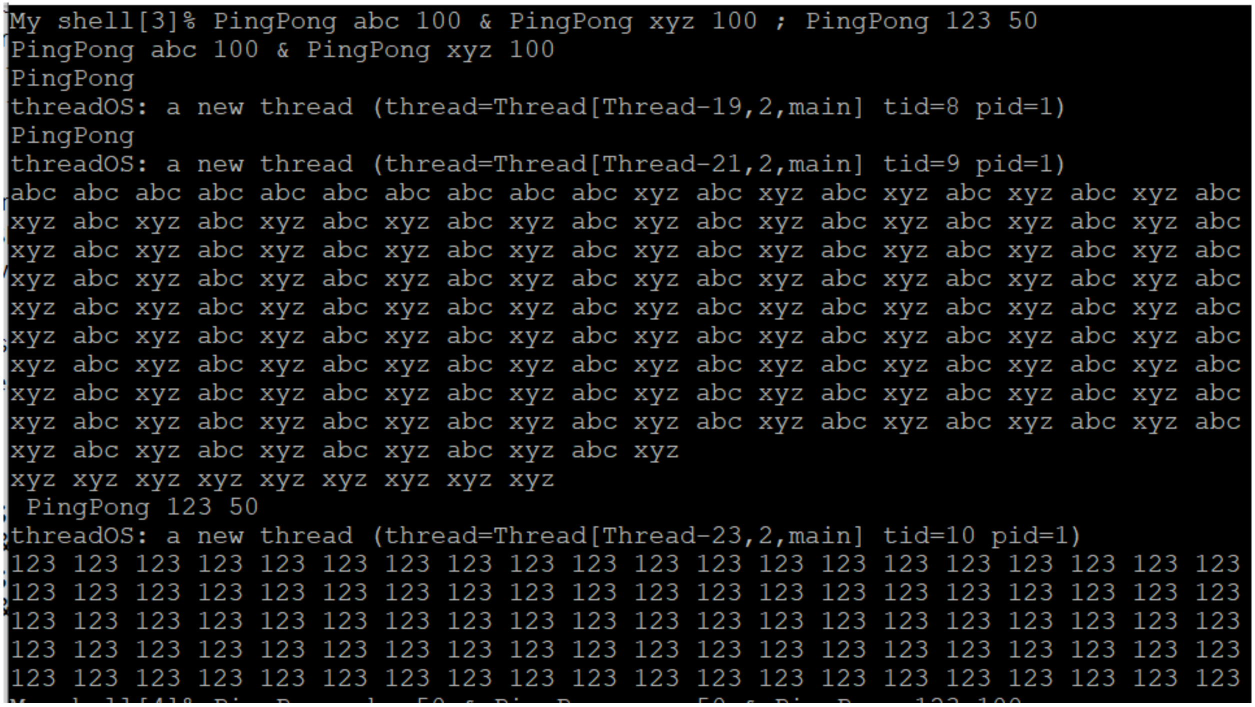
```
My shell[1]% PingPong abc 100 ; PingPong xyz 50 ; PingPong 123 100
PingPong
threadOS: a new thread (thread=Thread[Thread-7,2,main] tid=2 pid=1)
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc 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 abc
PingPong
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 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 xyz
PingPong
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
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
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
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
```

My Shell[2]% Output:



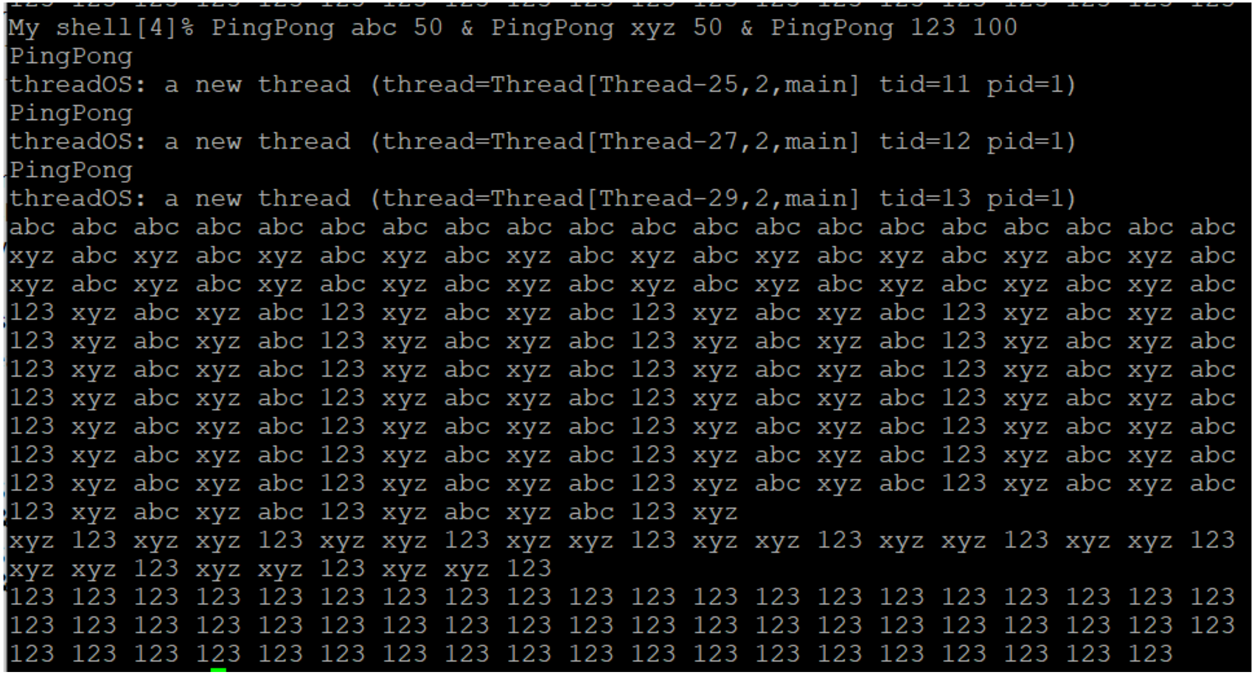
```
My shell[2]% PingPong abc 50 ; PingPong xyz 100 & PingPong 123 100
PingPong abc 50
threadOS: a new thread (thread=Thread[Thread-13,2,main] tid=5 pid=1)
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc
abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc 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 abc
 PingPong xyz 100 & PingPong 123 100
PingPong
threadOS: a new thread (thread=Thread[Thread-15,2,main] tid=6 pid=1)
PingPong
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 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 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 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 123 xyz 123
123 123 123 123 123 123 123 123 123
```

My Shell[3]% Output:



```
My shell[3]% PingPong abc 100 & PingPong xyz 100 ; PingPong 123 50
PingPong abc 100 & PingPong xyz 100
PingPong
threadOS: a new thread (thread=Thread[Thread-19,2,main] tid=8 pid=1)
PingPong
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 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 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 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
xyz xyz xyz xyz xyz xyz xyz xyz xyz
 PingPong 123 50
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
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
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
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
```

My Shell[4]% Output:



```
My shell[4]% PingPong abc 50 & PingPong xyz 50 & PingPong 123 100
PingPong
threadOS: a new thread (thread=Thread[Thread-25,2,main] tid=11 pid=1)
PingPong
threadOS: a new thread (thread=Thread[Thread-27,2,main] tid=12 pid=1)
PingPong
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 abc
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 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 abc xyz abc 123 xyz abc 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
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123
```