Ethan Crawford

CSS 430

10/7/2008

Program 1

# Part 1: processes.cpp

## Algorithm

The program creates pipe 1 and forks. The parent process waits for the child process to complete. The child process creates pipe 2 and forks. The great-grandchild runs ‘ps –A’, piping the output to the grandchild, which uses the file descriptor to feed STDIN for ‘grep argv[1]’, emitting the output to STDOUT which has been redirected to pipe 1. The parent uses the pipe1 file descriptor to feed STDIN for ‘wc –l’, emitting the final output (process count) to STDOUT. (See figure 1).

processes.exe

fork()

Parent:

wait();

wc –l;

Child:

fork()

Grandchild:

wait();

grep argv[1];

Great-grandchild:

ps -A

Pipe 2

Pipe 1

***Figure 1****: processes.exe structure*

## Tests:

* Invalid/null argv[1]
* Check for processes when >= 1 are running
* Check for processes when 0 are running

## Output:

[ethancr@uw1-320-31 processes]$ ./processes mingetty6

[ethancr@uw1-320-31 processes]$ ps -A | grep mingetty | wc -l6

[ethancr@uw1-320-31 processes]$ ./processes ksand0

[ethancr@uw1-320-31 processes]$ ps -A | grep kscand | wc -l0

[ethancr@uw1-320-31 processes]$ ./processes sendmail0

[ethancr@uw1-320-31 processes]$ ps -A | grep sendmail | wc -l0

## Source code:

See next page

# Part 2: Shell.java

## Algorithm:

Shell.java inherits from java.lang.Thread and implements a single run() method. This method contains an endless loop that waits for user input via StdLib.cin() . When input is received, the tokens are processed. Checks for end-of-token-array and special characters ‘&’ and ‘;’ are made.

When an entire token group has been detected, the tokens are copied to a new string array and passed to StdLib.exec(). If a wait has been specified, SysLib.join() is called until the desired process has exited.

## Tests:

* NULL command-prompt
* One command, no special trailing character
* one command, one special trailing character
* Two or more commands with special character separators

## Output:

[ethancr@uw1-320-31 ThreadOS]$ java Boot

threadOS ver 1.0:

Type ? for help

threadOS: a new thread (thread=Thread[Thread-3,2,main] tid=0 pid=-1)

-->l Shell

l Shell

shell[1]% PingPong abc 10000000 & PingPong xyz 10000000 & PingPong 123 10000000 &

threadOS: a new thread (thread=Thread[Thread-6,2,main] tid=1 pid=0)

threadOS: a new thread (thread=Thread[Thread-8,2,main] tid=2 pid=0)

threadOS: a new thread (thread=Thread[Thread-10,2,main] tid=3 pid=0)

shell[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 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 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 123 xyz 123 xyz xyz 123 xyz 123 xyz xyz 123 xyz 123 xyz xyz 123 xyz 123

123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123

PingPong abc 10000000 ; PingPong xyz 10000000 ; PingPong 123 10000000 ;

threadOS: a new thread (thread=Thread[Thread-12,2,main] tid=4 pid=0)

abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc abc 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-14,2,main] tid=5 pid=0)

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

threadOS: a new thread (thread=Thread[Thread-16,2,main] tid=6 pid=0)

123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123 123

shell[1]% exit

## Source code:

See next page