Yacine Manseur

10/2/2015

Operating Systems

Problem Set 3

**Sample Output:**

*Interactive case:*

Yacine@Yacine-Laptop ~/OS/OS/PS3

$ ./shell

ls -l >ls.out

Executing command ls with arguments "-l"

Child process 21064 returned with return code 0,

consuming 0.045000 real seconds, 0.030000 user, 0.015000 system

dir

Executing command dir

~$3Writeup.docx line2.out\r ls.out PS3Writeup.docx shell.c shell.exe.stackdump

line1.out\r line3.out\r ls.out\r sample.bash shell.exe

Child process 18720 returned with return code 0,

consuming 0.000000 real seconds, 0.000000 user, 0.000000 system

Yacine@Yacine-Laptop ~/OS/OS/PS3

$ cat ls.out

total 111

-rwxr-xr-x 1 Administrators None 162 Oct 3 17:49 ~$3Writeup.docx

-rw------- 1 Yacine None 21 Oct 3 17:46 line1.out

-rw------- 1 Yacine None 25 Oct 3 17:46 line2.out

-rw------- 1 Yacine None 20 Oct 3 17:46 line3.out

-rw------- 1 Yacine None 0 Oct 3 18:04 ls.out

-rw------- 1 Yacine None 516 Oct 3 17:46 ls.out

-rwxr-xr-x 1 Administrators None 25869 Oct 3 17:49 PS3Writeup.docx

-rwxr-xr-x 1 Administrators None 200 Oct 3 17:48 sample.bash

-rwxr-xr-x 1 Administrators None 4574 Oct 3 17:40 shell.c

-rwxr-xr-x 1 Yacine None 68816 Oct 3 17:29 shell.exe

-rwxr-xr-x 1 Yacine None 359 Oct 3 16:13 shell.exe.stackdump

*Interpreter case:*

Yacine@Yacine-Laptop ~/OS/OS/PS3

$ ./shell

cat sample.bash

#!./shell

echo Welcome to my Shell! >line1.out

echo This is my sample script >line2.out

echo I hope you like it! >line3.out

ls -l >ls.out

cat line1.out

cat line2.out

cat line3.out

cat ls.out

Child process 21288 returned with return code 0,

consuming 0.046000 real seconds, 0.000000 user, 0.046000 system

./sample.bash

Executing command ./sample.bash

Executing command echo with arguments "Welcome" "to" "my" "Shell!"

Child process 19896 returned with return code 0,

consuming 0.046000 real seconds, 0.015000 user, 0.031000 system

Executing command echo with arguments "This" "is" "my" "sample" "script"

Child process 19916 returned with return code 0,

consuming 0.000000 real seconds, 0.000000 user, 0.000000 system

Executing command echo with arguments "I" "hope" "you" "like" "it!"

Child process 17412 returned with return code 0,

consuming 0.000000 real seconds, 0.000000 user, 0.000000 system

Executing command ls with arguments "-l"

Child process 21368 returned with return code 0,

consuming 0.077000 real seconds, 0.000000 user, 0.077000 system

Welcome to my Shell!

Child process 18180 returned with return code 0,

consuming 0.046000 real seconds, 0.000000 user, 0.046000 system

This is my sample script

Child process 20544 returned with return code 0,

consuming 0.046000 real seconds, 0.000000 user, 0.046000 system

I hope you like it!

Child process 21360 returned with return code 0,

consuming 0.030000 real seconds, 0.000000 user, 0.030000 system

total 111

-rwxr-xr-x 1 Administrators None 162 Oct 3 17:49 ~$3Writeup.docx

-rw------- 1 Yacine None 21 Oct 3 18:06 line1.out

-rw------- 1 Yacine None 25 Oct 3 18:06 line2.out

-rw------- 1 Yacine None 20 Oct 3 18:06 line3.out

-rw------- 1 Yacine None 712 Oct 3 18:04 ls.out

-rw------- 1 Yacine None 0 Oct 3 18:06 ls.out

-rwxr-xr-x 1 Administrators None 25869 Oct 3 17:49 PS3Writeup.docx

-rwxr-xr-x 1 Administrators None 200 Oct 3 17:48 sample.bash

-rwxr-xr-x 1 Administrators None 4574 Oct 3 17:40 shell.c

-rwxr-xr-x 1 Yacine None 68816 Oct 3 17:29 shell.exe

-rwxr-xr-x 1 Yacine None 359 Oct 3 16:13 shell.exe.stackdump

Child process 20704 returned with return code 0,

consuming 0.045000 real seconds, 0.015000 user, 0.030000 system

Child process 21228 returned with return code 0,

consuming 0.367000 real seconds, 0.030000 user, 0.337000 system

**Sample.bash:**

#!./shell

echo Welcome to my Shell! >line1.out

echo This is my sample script >line2.out

echo I hope you like it! >line3.out

ls -l >ls.out

cat line1.out

cat line2.out

cat line3.out

cat ls.out

**Appendix:**

// Yacine Manseur

// Cooper Union Fall 2015

// ECE 357: Operating Systems

// Problem Set 2

// shell.c

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/time.h>

#include <sys/resource.h>

#include <fcntl.h>

#include <errno.h>

#include <sys/wait.h>

#include <sys/stat.h>

char\* readLines(FILE \*fp)

{

char\* buffer = (char \*)malloc(sizeof(char) \* 128);

if (buffer == NULL)

{

fprintf(stderr, "Error allocating memory.\n");

exit(1);

}

int c = getc(fp);

int i = 0;

while(c != '\n' && c != EOF)

{

buffer[i] = c;

i++;

c=getc(fp);

}

if (c == EOF)

exit(0);

buffer[i] = '\0';

realloc(buffer, i + 1);

return buffer;

}

int main(int argc, char \*argv[])

{

FILE \*fp;

int ii;

if (argc > 1)

{

// Open given file for reading

fp = fopen(argv[1], "r");

if(fp == NULL)

{

fprintf(stderr, "Error reading file: %s\n", strerror(errno));

exit(1);

}

}

else

{

// Set fp to stdin if nothing specified

fp = stdin;

}

char \*line;

while ((line = readLines(fp)) != NULL && !feof(fp))

{

// Line is a comment. Ignore it.

if(line[0] == '#')

continue;

char \*word;

char \*\*args = NULL;

int numSpaces = 0;

word = strtok(line, " ");

while(word != NULL)

{

numSpaces++;

args = realloc(args, sizeof(char\*) \* numSpaces);

if(args == NULL)

{

fprintf(stderr, "Error allocating memory.\n");

exit(1);

}

args[numSpaces-1] = word;

word = strtok(NULL, " ");

}

//Set end of args to null

args = realloc(args, sizeof(char\*) \* (numSpaces+1));

args[numSpaces] = NULL;

pid\_t pid;

struct rusage ru;

int status;

int flag = 0;;

switch (pid=fork())

{

case -1:

perror("Fork failed.");

exit(1);

break;

case 0:

if(numSpaces > 1)

{

char\* path = NULL;

int oldfd, newfd;

if(strstr(args[numSpaces-1], "<"))

{

path = strstr(args[numSpaces-1], "<");

path++;

oldfd = open(path, O\_RDONLY);

// redirect to stdin

newfd = 0;

flag = -1;

}

else

{

if(strstr(args[numSpaces-1], "2>>"))

{

path = strstr(args[numSpaces-1], "2>>");

path++;

path++;

path++;

oldfd = open(path, O\_WRONLY | O\_CREAT | O\_APPEND, S\_IREAD | S\_IWRITE);

// redirect to stderr

newfd = 2;

flag = -1;

}

else if(strstr(args[numSpaces-1], ">>"))

{

path = strstr(args[numSpaces-1], ">>");

path++;

path++;

oldfd = open(path, O\_WRONLY | O\_CREAT | O\_APPEND, S\_IREAD | S\_IWRITE);

// redirect to stdout

newfd = 1;

flag = -1;

}

else if(strstr(args[numSpaces-1], "2>"))

{

path = strstr(args[numSpaces-1], "2>");

path++;

path++;

oldfd = open(path, O\_WRONLY | O\_CREAT | O\_TRUNC, S\_IREAD | S\_IWRITE);

// redirect to stderr

newfd = 2;

flag = -1;

}

else if(strstr(args[numSpaces-1], ">"))

{

path = strstr(args[numSpaces-1], ">");

path++;

oldfd = open(path, O\_WRONLY | O\_CREAT | O\_TRUNC, S\_IREAD | S\_IWRITE);

// redirect to stdout

newfd = 1;

flag = -1;

}

}

if (oldfd < 0 || newfd < 0)

{

fprintf(stderr, "Error opening file: %s\n", strerror(errno));

exit(1);

}

else

{

dup2(oldfd, newfd);

close(oldfd);

if (flag == -1)

args[numSpaces-1] = '\0';

}

}

fprintf(stderr, "Executing command %s", args[0]);

if(numSpaces != 1)

fprintf(stderr, " with arguments ");

for(ii = 1; args[ii] != NULL; ii++)

fprintf(stderr, "\"%s\" ", args[ii]);

fprintf(stderr, "\n");

if(execvp(args[0], args) == -1)

{

fprintf(stderr, "Error executing file: %s\n", strerror(errno));

exit(1);

}

break;

default:

if(wait3(&status,0,&ru) == -1)

{

perror("wait3");

}

else

{

fprintf(stderr, "Child process %d returned with return code %d,\nconsuming %ld.%.6d real seconds, %ld.%.6d user, %ld.%.6d system\n",

pid,

status,

ru.ru\_utime.tv\_sec + ru.ru\_stime.tv\_sec,

ru.ru\_utime.tv\_usec + ru.ru\_stime.tv\_usec,

ru.ru\_utime.tv\_sec,

ru.ru\_utime.tv\_usec,

ru.ru\_stime.tv\_sec,

ru.ru\_stime.tv\_usec);

}

break;

free(line);

free(args);

}

}

exit(0);

}