University of Southern
Maine

M. FENTON LIBBY
COS 350
4 MAY 2021

Program 5 Write-Up

Systems Programming

Summa

From the starting code, only the file "execute.c" was modified to provide the desired functionality, the modified code follows:

```
       /* execute.c - code used by small shell to execute commands */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <dirent.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <pwd.h>

int cd ( char* pth );
int isin ( int num , int array[] );
void append ( int array[] , int new_member );
int bgjobs = 0;
int bgpids[10] = {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};


int execute ( char* argv[] )
/*
 * purpose: run a program passing it arguments
 * returns: status returned via wait, or -1 on error
 *  errors: -1 on fork() or wait() errors
 */
{
  int   pid ;
  int   child_info = -1;
  int bg = 0;
  int argc = 1;
  if ( argv[0] == NULL )     /* nothing succeeds */
    return 0;

  while(argv[argc] != NULL){
    argc++;
  }

  if (argv [ 1 ] != NULL & strcmp( argv [ argc - 1 ] , "&" ) == 0 ){
    bg = 1;
    argv[ argc - 1 ] = NULL;
  }

  if ( strcmp ( argv[0] , "exit" ) == 0 ) {/* Special Case: Exit [value]*/

    printf ( "Exiting with status %d\n" , ( argv [ 1 ] == NULL ) ? 0 : atoi ( argv [ 1 ]
        ) );
    exit ( ( argv[1] == NULL ) ? 0 : atoi ( argv [ 1 ] ) );

  }
```

```
  else if ( strcmp ( argv[0] , "cd" ) == 0 ) { /* Special Case: cd [dir]*/

    cd(argv[1]);

  }
  //GENERAL CASE
  else if ( (pid = fork()) == -1 ) /* Create fork, perror if it fails*/
    perror("fork");
  else if ( pid == 0 ){
    signal(SIGINT, SIG_DFL);
    signal(SIGQUIT, SIG_DFL);
    execvp(argv[0], argv);
    perror("cannot execute command");
    exit(1);
  }
  else {

    if ( waitpid ( -1 , &child_info , bg ? WNOHANG : 0 ) == -1 )
      perror("wait");
  }

  int temp;
  if ((temp = isin ( pid , bgpids )) != -1){
    bgjobs--;
    printf("[%d]+ Complete %d\n", temp, bgpids[temp]);
    bgpids[temp]=-1;
  }

  if ( bg ) {
    bgjobs++;
    printf("[%d] %d\n" , bgjobs ,pid);
    append(bgpids, pid);
  }
  return child_info;
}


int cd ( char* pth ){
  char path[200];
  /* Handle base case (cd to user home), i.e. pth==NULL*/
  if (pth == NULL){
    char* home;
    home = getpwuid( getuid ( ) ) -> pw_dir ;
    chdir( home );

  } else {
  strcpy(path,pth);
  /* identify if path is absolute or relative */
  if(pth[0] != '/') {//if it is relative, find cwd and use it for chdir
      char cwd[200];
      getcwd ( cwd , sizeof ( cwd ) );
      strcat ( cwd , "/" );
      strcat ( cwd , path );
      chdir ( cwd );

  } else {//if it is absolute, go there

      chdir(pth);
  }
```

```
  }
  return 0;
}


int isin(int num, int array[]){

  for ( int i = 0 ; i < 10 ; i++ ){
    if ( array [ i ] == num ){
      return i;
    }
  }
  return -1;
}

void append ( int array[] , int new_member ){

  for ( int i = 0 ; i < 10 ; i++ ){

    if (array[i] == -1){
      array[i] = new_member;
      return;
    }

  }

}
```

The makefile was also modified to reflect name changes:

```
CC=gcc
CFLAGS=-g

mysh: mysh.o splitline.o execute.o
$(CC) $(CFLAGS) -o mysh mysh.o splitline.o execute.o
```

A log of the required testing results:

```
Script started on 2021-05-04 08:04:54-0400
mflibby@honeybee:~/cos350/prog/prog5$ mysh
--> ls
execute.c execute.o Makefile mysh mysh.c mysh.h mysh.o smsh1 smsh1.c smsh.h splitline.c
    splitline.o typescript
--> date
Tue May 4 08:05:05 EDT 2021
--> bogus
cannot execute command: No such file or directory
--> exit
Exiting with status 0
mflibby@honeybee:~/cos350/prog/prog5$ mysh
--> exit 7
Exiting with status 7
mflibby@honeybee:~/cos350/prog/prog5$ echo $?
7
mflibby@honeybee:~/cos350/prog/prog5$ mysh
--> pwd
/home/students/mflibby/cos350/prog/prog5
--> cd ..
--> pwd
/home/students/mflibby/cos350/prog
--> cd /usr/bin
--> pwd
/usr/bin
--> cd bogus
--> pwd
/usr/bin
--> cd
--> pwd
/home/students/mflibby
--> sleep 10 &
[1] 21960
--> ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN TTY        TIME CMD
0 S 21925 20330 20328 0 80   0 -  3615 wait  pts/9    00:00:00 bash
0 S 21925 21022 20330 0 80   0 -  1183 wait  pts/9    00:00:00 mysh
0 S 21925 21960 21022 0 80   0 -  1869 hrtime pts/9  00:00:00 sleep
0 R 21925 22007 21022 0 80   0 -  7230 -     pts/9    00:00:00 ps
--> ps -l
--> F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN TTY        TIME CMD
0 S 21925 20330 20328 0 80   0 -  3615 wait  pts/9    00:00:00 bash
0 S 21925 21022 20330 0 80   0 -  1183 wait_w pts/9  00:00:00 mysh
0 R 21925 22500 21022 0 80   0 -  7230 -     pts/9    00:00:00 ps

--> ps -l
--> F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN TTY        TIME CMD
0 S 21925 20330 20328 0 80   0 -  3615 wait  pts/9    00:00:00 bash
0 S 21925 21022 20330 0 80   0 -  1183 wait_w pts/9  00:00:00 mysh
0 R 21925 22710 21022 0 80   0 -  7230 -     pts/9    00:00:00 ps

--> exit
Exiting with status 0
mflibby@honeybee:~/cos350/prog/prog5$ exit
exit

Script done on 2021-05-04 08:07:13-0400
```

There are still a few problems with this code; namely, the completion of background tasks does not currently notify that user - more investigation is needed to ascertain the best way to do this -, also, after starting a background task, a formatting error occasionally occurs, where in the prompt "–>" is printed before the output of a command, and so the output of that command is printed after the prompt, leaving the next line without a prompt - hitting return seems to temporarily fix this, but the formatting error remains until the program is restarted.