

Programming Assignment 1  
Introduction to OS Calls  
Due: February 8, 2013, 23:00 (UTA time)

Description:

You will develop a small "system shell utility" ("shu") for Unix (or similar OS). A common complaint of the most-used UNIX shells (Bourne, Korn, C) is that it is difficult to remember (long) file names, and type "long" command names. A "menu" type shell allows a user to "pick" an item (file or command) from a "menu".

You will display a simple menu, the following is a suggestion:

Current Working Dir: /home/os.labs/IAMMe  
It is now: 8 February 2013, 10:58 PM

Files:           0. a.c  
                 1. a.out  
                 2. a.txt  
                 3. assignment1  
                 4. program

Directories:    0. ..  
                 1. my\_dir

Operation:      E Edit  
                 R Run  
                 C Change Directory  
                 Q Quit

You will read a single key "command" entered by the user, then a file or directory number and shu will output some system information on the terminal, or run a program (by means of systems calls).

The commands you will implement:

  Edit - opens a text editor with a file.

  Run - runs an executable program. You should handle parameters.

  Change - changes working directory.

You may want to also implement:

  Prev and Next give the next "N" (or prev.) files if not all names fit on one screen. Or similar.

If the "shu" program has an argument, use it as the directory starting point (working dir), like: "./shu /bin".

You should provide a reasonable user interface. (See below)

You may write this in ADA, Assembler, C, C++ or Java. (Others with permission)

You should target shu to Unix, MacOS, (or similar).

Constraints:

  How many files and directories can you handle (max 1024)

  How long is a file name (max 2048 characters, really: limits.h, NAME\_MAX)

Bonus:

  Add "sort options" (by name, date, size, extension, etc.)

  Add a pull-down menu using terminal codes or cursers.

**Please, Email ONLY to the class account. All work must be your own.**

You must e-mail this lab, working (or partially) by the due date. The e-mail subject should clearly state the lab number.

You may (optionally) demonstrate this lab, working (or partially) to the

GTA before the due date.

Your program should be well commented and documented, make sure the first few lines of your program contain your name, this course number, and the lab number.

Your comments should reflect your design and issues in your implementation.

Your design and implementation should address error conditions:

what if an illegal command is entered or misspelled, what if a path can not be reached, or an "executable file" is not executable.

Hints: (you may use this code)

```
/* Some example code and prototype -
contains many, many problems: should check for return values
(especially system calls), handle errors, not use fixed paths,
handle parameters, put comments, watch out for buffer overflows,
security problems, use environment variables, etc.
*/

#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <dirent.h>
#include <string.h>
#include <time.h>

int main(void) {
    pid_t child;
    DIR * d;
    struct dirent * de;
    int i, c;
    char s[256], cmd[256];
    time_t t;

    while (1) {
        getcwd(s, 200);
        printf( "\nCurrent Directory: %s \n", s);

        d = opendir( "." );
        c = 0;
        while ((de = readdir(d))){
            printf( " ( %d %s %s ) ", c++, de->d_name,
                ((de->d_type) & DT_DIR) ? "Directory": "File" );
        }
        closedir( d );
        printf( "\n" );

        c = getchar( );
        switch (c) {
            case 'q': exit(0); /* quit */
            case 'e': printf( "Edit what?:" );
                scanf( "%s", s );
                strcpy( cmd, "pico " );
                strcat( cmd, s );
                system( cmd );
                break;
            case 'r': printf( "Run what?:" );
                scanf( "%s", cmd );
                system( cmd );
                break;
            case 'c': printf( "Change To?:" );
                scanf( "%s", cmd );
                chdir( cmd );
                break;
            case 't': t = time( NULL );
                printf( "Time: %s\n", ctime( &t ));
                break;
        }
    }
}
```