Ficha Prática 5

Sistemas Operativos 2023

Esta ficha é dedicada a implementação da funcionalidade embutida no shell, quer dizer que o nosso interpretador de comandos não necessita de executar um comando externo para os comandos "calc" , "bits" e "isjpeg" e otimzar a função exececute()

1 Implementação da função "builtin" calc -uma calculadora - no Ficheiro calc.c

A função embutida no soshell, *calc*, efetuará a operação dum calculador simples e imprimirá o resultado no ecrã. Os operandos devem ser considerados ***floats*** e o operador é um dos seguintes caracter (+, -, *, / e exponenciação ^) com semântica habitual.

```
SoShell: prompt>  calc  operando1 operador operando2
```

Exemplo de funcionamento e *output*:

```
SOSHELL: prompt> calc 2.0 + 3.0
Resultado calc 5.000000
```

Para efetuar esta funcionalidade deverá criar uma nova função implementada num novo ficheiro "**calc.c**" que implemente a função do calculadora, esta função será depois chamada a partir da builtin().

void calc(char *value1, char *op, char *value2).

Vai precisar das funções: "string to float", **atof() e powf()** do **math.h.**

Nota: A sua implementação deverá tratar de erros como divisão por zero

```
if ( 0 == strcmp(args[0], "calc" e n° args é 3) {
    //chamar a função calc()
    return 1 ;  //comando embutido
}
```

2 Implementação da função "builtin" *bits* – calculadora de *bits* -- no Ficheiro calc.c

O objetivo desta tarefa é adicionar a funcionalidade **`bits`**, que efetuará a operação de uma calculadora simples para operadores binários de bits e imprime o resultado no ecrã.

```
SOSHELL: prompt> bits operando1 operador  operando2
```

Exemplo de funcionamento e *output*:

```
SOSHELL: prompt> bits 3 & 2
Resultado bits 2
```

Nota : Os operandos são inteiros e o operador binário será um caracter: **`&`**, **`^`** ou **`|`**

```
&  binary bitwise AND
^  binary bitwise exclusive OR
|  binary bitwise inclusive OR
```

Também pode considerar outros, como a negação **`~`** e os *shifts* << *e* >>.

Para adicionar esta funcionalidade, implementa a função **`void bits( char *op1, char *op, char *op2 )`**.

Esta função será depois chamada a partir da **`builtin()`**.

Vai provavelmente precisar da função: *string to integer* **`atoi()`**.

Considere a seguinte sugestão na resolução deste exercício:

```
if (0 == strcmp(args[0], "bits") e N°Args  é 3  (ou 2 se considerar ~)
    chamar a função bits()
    return 1 ;  //commando embutido
}
```

3 Implementação da função "builtin" *isjpeg()*

Para detetar se um ficheiro tem ou não o formato JPG investigarmos os primeiros "magic bytes": a função deverá ler os primeiros 4 bytes do ficheiro. Se for um ficheiro "JPG" a função devolverá 1.

```
int isjpg ( int fileDescriptor )  //esboço da função
{
   /* ficheiro tem que estar aberto */
    unsigned char b[4];
   int n = read(fileDescriptor,b,4);
    if ( n ) …
    //voltar ao início do ficheiro com lseek
   if ( b[0]==0xff && b[1]==0xd8 && b[2]==0xff && b[3]==0xe0)
      return 1;
   return 0;
}
```

0xFF-D8-FF-E0 — Standard JPEG/JFIF file
0xFF-D8-FF-E1 — Standard JPEG file with Exif metadata
0xFF-D8-FF-E2 — Canon Camera Image File Format (CIFF) JPEG file
0xFF-D8-FF-E8 — Still Picture Interchange File Format (SPIFF)

Ver : File signatures - magic header/trailer bytes- https://www.garykessler.net/library/file_sigs.html

Ficheiros de Teste: Obter com o programa "wget"
https://www.di.ubi.pt/~crocker/lena.jpg
https://upload.wikimedia.org/wikipedia/commons/thumb/2/28/JPG_Test.jpg/953px-JPG_Test.jpg
https://www.di.ubi.pt/~operativos/index_files/image003.jpg
Para visualizar a primeira linha do ficheiro com bytes hex : od -x file.jpg | head -1

4 Otimização da função execute()

Se o utilizador escrever um comando usando um caminho relativo ou absoluto (portanto o comando contêm um /) não é necessário chamar a função execvp que também pesquisa os comandos no PATH (na verdade também possa implementar este verificação) e podemos chamar diretamente a função execl() . Implemente este pequena mudança - quer dizer no *soshell* na função execute() faz esta verificação (usando strstr()) e depois chamar execvp() ou execv() conforme.

From the Linux Manual: "The **exec**() family of functions replaces the current process image with a new process image. The functions described in this manual page are front-ends for **execve** "
→quer dizer são apenas função de ajuda para fazer interface / chamar a verdadeira system call execve.

Avaliação

Pedo ao professor um ficheiro para testar a suas implementações