

SISTEMAS OPERATIVOS

Práctica 1. Llamadas al sistema operativo

Presentación

2

- ☐ Enunciado
- ☐ Llamadas utilizadas
- ☐ Aspectos clave

Práctica

3

- Realizar tres pequeños programas en C manipulen ficheros y directorios.
- Cada programa en un único fichero
 - *mycat.c* → Muestra por pantalla el contenido de un fichero
 - *myls.c* → Muestra las entradas de un directorio
 - *mysize.c* → Muestra el nombre y tamaño de las entradas de un directorio que son ficheros regulares.

mycat

4

- ❑ `./mycat <fichero de entrada>`
- ❑ Resultado: impresión por pantalla
 - ❑ El programa debe mostrar todo el contenido del fichero.
 - ❑ El programa debe devolver -1 si no se le ha pasado ningún argumento de entrada.
 - ❑ El programa debe devolver -1 si hubo un error al abrir el fichero (e.g. el fichero no existe).
- ❑ Ejemplo

```
usuario@laptop:~$ ./mycat p1_pruebas/f1.txt
```

| | | | | |
|---------|---|----|----------|----------|
| Nombre1 | V | 32 | 09834320 | 24500.34 |
| Nombre2 | M | 30 | 58854450 | 24700.55 |
| Nombre3 | V | 27 | 45678978 | 28000.0 |

myls

5

- ❑ `./mys <dir>` `//Lista dir`
- ❑ `./mys` `//Lista directorio actual`
- ❑ Resultado: listado de las entradas del directorio
 - ❑ Una entrada por línea
- ❑ Ejemplo

```
usuario@laptop:~$ ./mys p1_pruebas/  
dirC  
f1.txt  
dirA  
f2.txt  
.  
..
```

mysize

6

- ./mysize
- Resultado: listado de los ficheros regulares y sus tamaños.
 - El programa solo mostrará datos de ficheros regulares.
 - El programa mostrara los datos en el siguiente formato:
<nombre><tabulador><tamaño>.
- Ejemplo

```
usuario@laptop:~$ cd p1_pruebas/  
../mysize  
f1.txt      87  
f2.txt      87
```

Llamadas al sistema

7

- ❑ Para el programa mycat debéis utilizar las llamadas relativas a ficheros (open, read, write, y close).
- ❑ Para el programa myls debéis utilizar las llamadas relativas a directorios (getcwd, opendir, readdir y closedir).
- ❑ Para el programa mysize debéis mezclar ambos tipos.
- ❑ La documentación de dichas llamadas la podéis encontrar en el man (ver anexos de la práctica)

Gestión de los argumentos de entrada

8

- ❑ Extraer las rutas de los archivos y directorios pasados como argumentos
- ❑ Realizar control de errores
- ❑ Ejemplo de manipulación de argumentos:

```
int main( int argc, char *argv[]){  
    int retorno = 0;  
    if( argc >= 3){  
        printf(argv[0]);  
        printf(argv[1]);  
        printf(argv[2]);  
    }else{  
        printf("Argumentos insuficientes\n");  
        retorno = -1;  
    }  
    return retorno;  
}
```


Impresiones por pantalla

9

- mycat
 - Escribir directamente en la salida estándar el buffer que contenga los datos leídos.
 - `write(STDOUT_FILENO, buffer, TAM_BUFFER);`
- myls
 - Imprimir con la llamada *printf* el campo *d_name* de la estructura *dirent*.
 - `printf("%s\n", entrada->d_name);`
- mysize
 - Imprimir con la llamada *printf* el campo *d_name* de la estructura *dirent* y el tamaño obtenido, con tabulador entre medias.
 - `printf("%s\t%ld\n", entrada->d_name, tam);`

Código de apoyo

10

- Se os proporciona como código de partida una carpeta comprimida que ya incluye los ficheros de los programas creados y un fichero para compilarlos
- Para compilar: situarse en la carpeta y teclear el comando make
-

Corrector

11

- Se os proporciona un corrector en Python que garantiza que se cumple:
 - Formato de entrega (zip)
 - Nombres de programas
 - Compilación
 - Funcionalidad **básica**
- Para ejecutar:
 - *python corrector_ssoo_p1_2019.py <file .zip>*
-

Corrector

12

- Se os proporciona un corrector en Python que garantiza que se cumple:
 - Formato de entrega (zip)
 - Nombres de programas
 - Compilación
 - Funcionalidad **básica**
- Para ejecutar:
 - *python corrector_ssoo_p1_2019.py <file .zip>*
-