



Área de Arquitectura y Tecnología de Computadores (ARCOS)

Universidad Carlos III de Madrid

SISTEMAS OPERATIVOS

Práctica 1. Llamadas al Sistema Operativo

Grado de Ingeniería en Informática
Grado en Matemática Aplicada y Computación
Doble Grado en Ingeniería Informática y Administración de
Empresas

Curso 2022-2023

- 1 Descripción de la Práctica
- 2 Entrega
- 3 Llamadas al Sistema

1 Descripción de la Práctica

2 Entrega

3 Llamadas al Sistema

Descripción

- Realizar tres pequeños programas en C que manipulen ficheros y directorios.
- Cada programa en un único fichero
 - **mywc.c** → Cuenta el número de líneas, palabras y bytes de un fichero
 - **myenv.c** → Busca una variable de entorno específica
 - **myls.c** → Muestra las entradas de un directorio

mywc

- `./mywc <fichero_de_entrada>`
- Resultado:
 - El programa debe mostrar el **número de líneas, palabras y bytes por consola, seguidos del nombre del fichero leído**
 - El programa debe devolver -1 en caso de error
 - El programa debe devolver 0 si todo funcionó correctamente
- Ejemplo

```
1      $ ./mywc p1_tests/f1.txt
2      2 15 85 p1_tests/f1.txt
```

myenv

- `./myenv <env> <fichero_salida>`
- Resultado:
 - El programa debe escribir **todas las entradas** con la variable, en el orden en que estén en el archivo de entrada (`env.txt`) **situado en el mismo directorio que myenv**
 - El programa debe separar cada entrada por **un salto de línea**
 - El programa debe crear el fichero `fichero_salida` vacío si no encuentra ninguna línea con la variable
 - El programa debe devolver -1 si se produce un error
 - El programa debe devolver 0 si todo funcionó correctamente
- Ejemplo:

```
1 $ ./myenv PATH ./getenv.out
2 PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games
```

mys

- `./mys <dir> //Lista dir`
- `./mys //Lista directorio actual`
- Resultado:
 - Listado de las entradas del directorio. **Una entrada por línea**
 - El programa debe devolver -1 si hubo un error al abrir el directorio (e.g. el directorio no existe)
- Ejemplo:

```
1      $ ./mys p1_tests/  
2      dirC  
3      f1.txt  
4      dirA  
5      f2.txt  
6      .  
7      ..
```

Código Fuente de Apoyo

- Se proporciona como código de partida una carpeta comprimida que incluye los ficheros de los programas a realizar (*mywc.c*, *myenv.c* y *myls.c*) y un fichero para compilarlos (*Makefile*)
- Para compilar: situarse en la carpeta y teclear el comando `make`

Probador

- Se proporciona a los alumnos el script en **python (Versión 3)** **probador_ssoo_p1.py**
- El probador deberá ejecutarse en las máquinas virtuales con Ubuntu Linux o en la Aulas Virtuales del Laboratorio de Informática
- El comando para ejecutar el corrector es el siguiente:

```
python3 probador_ssoo_p1.py <entregable.zip>
```

- Ejemplo:

```
$ python3 probador_ssoo_p1.py  
ssoo_p1_100254896_100047014.zip
```

- 1 Descripción de la Práctica
- 2 Entrega**
- 3 Llamadas al Sistema

Plazo de entrega y grupos

- Grupos de 3 personas máximo.
- Entrega de:
 - Código Fuente en un archivo comprimido
 - Memoria de práctica en PDF a través de entregador TURNITIN
 - Solamente podrá entregar un integrante del grupo
- Fecha de entrega:

12 de Marzo de 2023 (hasta las 23:55h)

- 1 Descripción de la Práctica
- 2 Entrega
- 3 Llamadas al Sistema**

Llamadas al Sistema

- Para el programa **mywc** debéis utilizar las llamadas relativas a ficheros (open, read, write, y close).
- Para el programa **myenv** debéis utilizar las llamadas relativas a ficheros (open, read, write, y close).
- Para el programa **mysls** debéis utilizar las llamadas relativas a directorios (getcwd, opendir, readdir y closedir).
- La documentación de dichas llamadas la podéis encontrar en el man (ver anexos de la práctica).

Gestión de los argumentos de entrada

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

```
1      int main (int argc, char *argv[]){  
2          int retorno = 0;  
3          if (argc >= 3){  
4              printf(argv[0]);  
5              printf(argv[1]);  
6              printf(argv[2]);  
7          }else{  
8              printf('Argumentos insuficientes\n');  
9              retorno = -1;  
10         }  
11         return retorno;  
12     }
```

Impresiones por pantalla

■ mywc

- Imprimir con la llamada printf el número de líneas, palabras, bytes y el nombre del fichero

```
printf("%d%d%d%s\n", lineas, palabras, bytes,  
fichero);
```

■ myenv

- Imprimir con la llamada printf cada una de las entradas coincidentes

```
printf("%s\n", entrada);
```

■ myls

- Imprimir con la llamada printf el campo *d_name* de la estructura *dirent*.

```
printf("%s\n", entrada->d_name);
```



Área de Arquitectura y Tecnología de Computadores (ARCOS)

Universidad Carlos III de Madrid

SISTEMAS OPERATIVOS

Práctica 1. Llamadas al Sistema Operativo

Grado de Ingeniería en Informática
Grado en Matemática Aplicada y Computación
Doble Grado en Ingeniería Informática y Administración de
Empresas

Curso 2022-2023