

Ejercicios propuestos

Tema 3 – Programación en C

Sistemas Operativos

- 1- Escriba un programa en C llamado “myseq” que haga lo mismo que el comando *seq(1)* sin opciones. Es decir, sólo tiene que entender:

myseq LAST

myseq FIRST LAST

myseq FIRST INCREMENT LAST

Utilice *atoi(3)* para convertir un string que representa un número entero a su valor entero.

Ejemplos de ejecución:

```
$ ./myseq 3
```

1

2

3

```
$ ./myseq 2 5
```

2

3

4

5

```
$ ./myseq 3 4 12
```

3

7

11

- 2- Escriba un programa en C llamado “invert” que muestre todos los argumentos que se le pasan al revés: primero la última palabra, al final la primera, y de cada palabra empezando por la última letra y terminando por la primera.

Ejemplo:

```
$ ./invert es cara la caca
```

```
acacal arace
```

- 3- Escriba un programa en C llamado “mycd” que cambie el directorio actual. Dicho programa debe permitir tanto el acceso mediante rutas absolutas como relativas, además de la posibilidad de acceder al directorio especificado en la variable HOME si no recibe ningún argumento, escribiendo la ruta absoluta del nuevo directorio actual de trabajo.

Ejemplo:

```
$ ./mycd /home/user/directorio
```

```
/home/user/directorio
```

```
$ ./mycd
```

```
/home/user/
```

Algunas llamadas al sistema y funciones necesarias: *chdir(2)*, *getenv(3)*, *getcwd(3)*

- 4- Escriba un programa en C llamado “findCommand” que busque en todos los directorios contenidos en la variable de entorno \$PATH el nombre de un mandato que se le pasa como parámetro comprobando que es ejecutable.

Algunas llamadas al sistema y funciones necesarias: *access(2)*, *getenv(3)*, *strcat(3)*

- 5- Escriba un programa en C llamado “myhead” que se comporte como *head(1)* leyendo de la entrada estándar, es decir, que muestra las 10 primeras líneas recibidas por la entrada estándar, o las N primeras si se llama al programa con “myhead -N”.

Se debe usar la función *fgets(3)* pasándole como tercer argumento la constante *stdin*, de una forma similar a:

```
char buf[1024];
```

```
fgets(buf, 1024, stdin);
```

- 6- Escriba un programa en C llamado “mytail” que se comporte como *tail(1)* leyendo de la entrada estándar, es decir, que muestra las 10 últimas líneas recibidas por la entrada estándar, o las N últimas si se llama al programa con “mytail -N”.

Algunas llamadas al sistema y funciones necesarias: *fgets(3)*, *strcpy(3)*