SISTEMAS OPERATIVOS - LABORATORIO
13 de Junio de 2023

Nombre _____ DNI _____

Apellidos _____ Grupo _____

ADVERTENCIA: Si el código no compila o su ejecución produce un error grave la puntuación de ese apartado será 0.

Cuestión 1. (5 puntos) Crear un programa, a partir de la plantilla suministrada en el examen, que:

a. **(1 punto)** Acepte un parámetro opcional -R procesado con **getopt**. Por defecto la opción estará desactivada.
b. **(1 punto)** Cuando no se pasa la opción -R el programa se comportará como el comando "ls -1f", que muestra por su salida estándar el nombre de todas las entradas del directorio que se indica en la línea de comandos. Si el directorio se omite se usa el directorio actual. Cada entrada del directorio se mostrará en una línea diferente.

```
$ mkdir -p a/aa
$ echo "Hola mundo" | tee a/a.txt  a/aa/aa.txt > /dev/null
$ ./ej1
a
.
..
ej1.c
Makefile
ej1
ej1.o
```

```
$ ls -1f
a
.
..
ej1.c
Makefile
ej1
ej1.o
```

c. **(3 puntos)** Cuando el programa recibe la opción -R mostrará el contenido recursivo, como el comando ls -1fR. Para ello, realizará las siguientes tareas:
    1. Mostrará el nombre del directorio a procesar seguido de ":"
    2. Mostrará el contenido del directorio a procesar (función del apartado anterior)
    3. Recorrerá el directorio a procesar y por cada subdirectorio creará un proceso, que debe ejecutar el mismo programa pero pasando como argumento la ruta al subdirectorio, que se forma concatenando el nombre del directorio con el del subdirectorio separados por "/".
    4. Antes de proseguir con la siguiente entrada esperará a que finalice el proceso creado.

```
$ ./ej1 -R
.:
a
.
..
ej1.c
Makefile
ej1
ej1.o

./a:
.
aa
..
a.txt

./a/aa:
.
..
aa.txt
```

```
$ ls -1fR
.:
a
.
..
ej1.c
Makefile
ej1
ej1.o

./a:
.
aa
..
a.txt

./a/aa:
.
..
aa.txt
```

Cuestion 2. Crea un programa que simule con hilos POSIX la sincronización entre los miembros de un equipo de atletismo en una carrera de relevos. Usaremos un hilo distinto para simular a cada corredor. La creación del hilo simula la llegada de uno de los corredores al estadio, obteniendo un dorsal (id que corresponde al orden de creación de los hilos). Al llegar, cada corredor debe ir primero a cambiarse, luego reservar su posición de salida en la carrera, moverse a la posición de salida que le corresponde y esperar a que todos los corredores se hayan colocado antes de que empiece la carrera. Una vez la carrera ha comenzado los corredores deben esperar a que sea su turno para correr, después correr y finalmente avisar al siguiente corredor de que es su turno.
La función principal de los hilos se corresponde con:

```
void *thmain(void *arg)
{
	int id = (int) (long long) arg;
	int pos;

	change_clothes(id);
	pos = get_position(id);
	move_position(id, pos);
	wait_all_ready(id);
	wait_my_turn(id, pos);
	run(id, pos);
	pass_on_relay(id);
	return NULL;
}
```

Con el enunciado de examen se proporciona un esqueleto del programa con algunas funciones implementadas. El alumno debe:

a. **(1 punto)** Crear el programa principal responsable de crear NTH (10) hilos que ejecuten la función thmain y esperar a que terminen, así como de declarar e inicializar los recursos y variables necesarias.
b. **(1 punto)** Implementar la función get_position, que permite a los corredores ir reservando una posición en la carrera de relevos según salen del vestuario.
c. **(1 punto)** Implementar la función wait_all_ready, una barrera para que la carrera no empiece hasta que todos los corredores estén en su posición listos.
d. **(1 punto)** Implementar la función wait_my_turn, que hace que cada corredor espere a que sea su turno.
e. **(1 punto)** Implementar la función pass_on_relay, que permite al corredor señalizar el turno del siguiente corredor.

El resto de funciones se dan implementadas en la plantilla.

Ejemplo:

```
$ ./ej2
Runner 0 is changing clothes
Runner 6 is changing clothes
Runner 5 is changing clothes
Runner 3 is changing clothes
Runner 4 is changing clothes
Runner 2 is changing clothes
Runner 1 is changing clothes
Runner 7 is changing clothes
Runner 8 is changing clothes
Runner 9 is changing clothes
Runner 5 reaches its starting position 0
Runner 8 reaches its starting position 1
Runner 7 reaches its starting position 2
Runner 2 reaches its starting position 7
Runner 0 reaches its starting position 3
Runner 3 reaches its starting position 4
Runner 1 reaches its starting position 6
Runner 6 reaches its starting position 5
Runner 4 reaches its starting position 9
Runner 9 reaches its starting position 8
```

```
Runner 5 running from position 0
Runner 8 running from position 1
Runner 7 running from position 2
Runner 0 running from position 3
Runner 3 running from position 4
Runner 6 running from position 5
Runner 1 running from position 6
Runner 2 running from position 7
Runner 9 running from position 8
Runner 4 running from position 9
```