



SISTEMAS OPERATIVOS - LABORATORIO
16 de Enero de 2024, Turno 1

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) Desarrollar un programa que implemente un intérprete de comandos muy simple. El programa consistirá en dos funciones: **main** y **setargs**. La función **setargs** se desarrollará en el apartado a y la función **main** en el apartado b. Se proporciona un esqueleto de código con un makefile para cada uno de los apartados, así como alguna ayuda para comprobar el correcto funcionamiento del ejercicio.

- a. **(2 puntos)** Implementar la función **setargs**. Esta función recibe dos argumentos: **args**, que es una cadena de caracteres con una línea de órdenes para el intérprete (ejemplo: "ls -l -t"), y **argv**, un puntero doble a char. La función devuelve un valor entero, que indica el número de subcadenas separadas por espacios contenidas en la cadena **args** (en el ejemplo anterior sería 3). El comportamiento de la función es como sigue:
- Si la función se invoca con **argv = NULL**, entonces sólo contará el número de subcadenas separadas por espacios en blanco, y no modificará la cadena apuntada por **args**.
 - Si la función se invoca con **argv != NULL**, además de lo anterior la función sustituirá en **args** todos los espacios por '\0', y copiará en **argv[i]** la dirección de comienzo de la i-ésima subcadena separada por espacios de **args**.
 - La función ignorará los espacios al comienzo de la cadena apuntada por **args**.

Se usará la función de la biblioteca estándar **isspace** para comprobar si un carácter es o no un espacio. Se proporciona un programa principal que permite comprobar el funcionamiento de la función **setargs**. Al ejecutar el programa veremos lo siguiente:

```
$ ./c1_a  
argv[0] = ls  
argv[1] = -l  
argv[2] = -t
```

- b. **(3 puntos)** Implementar el programa principal del intérprete (**main**), que se comportará del siguiente modo:
- El programa admite solo un parámetro obligatorio. El programa usará ese parámetro como el nombre de un fichero del que leer la entrada.
 - El programa leerá su entrada línea a línea hasta final de fichero, usando para ello la función **fgets**. Para cada línea se invocará primero **setargs** con el segundo parámetro a **NULL** para saber cuántas subcadenas hay separadas por espacios.
 - Con la información anterior se reservará espacio para un array tipo **argv** (**cargv**) correctamente formado (recordar que la última entrada debe ser **NULL**) y se volverá a invocar a **setargs** pasando **cargv** como segundo argumento.
 - Después el programa creará un nuevo proceso para que ejecute el programa indicado por esa línea (**cargv[0]** es el nombre del programa, el resto los argumentos). El programa esperará a que el nuevo proceso termine de ejecutar el comando antes de procesar la siguiente línea de la entrada.

Se proporciona un fichero de texto que puede ser usado para comprobar si el programa funciona correctamente:

```
$ ./c1_b test_c1b.txt  
test_c1b.txt c1_b c1_b.o Makefile c1_b.c  
SRC = $(wildcard *.c)  
TARGETS_SRC = $(wildcard c1_*.c)  
TARGETS = $(TARGETS_SRC:%.c=%)  
  
CC = gcc  
CFLAGS = -pthread -g $(CPPFLAGS)
```

```
LD_FLAGS = -pthread  
LIBS =  
  
all: $(TARGETS)  
ls -t  
head Makefile  
cat test_c1b.txt
```

Cuestión 2. (5 puntos) Modificar el programa de la práctica 5 (disco.c) para que admita un tercer tipo de cliente, que llamaremos cliente “special”, indicado en el fichero de entrada por el número 2 (0: cliente normal, 1: cliente vip, 2: cliente special). Debemos modificar las condiciones de sincronización del siguiente modo:

- Los clientes special no pueden entrar hasta que la discoteca se vacíe de clientes normales y clientes vip (puede haber otros clientes special).
- Los clientes special tienen la máxima prioridad, si hay algún cliente special esperando a entrar no pueden entrar clientes normales ni clientes vip.
- Los clientes special tienen acceso exclusivo a la discoteca, si hay algún cliente special en la discoteca los clientes normales y los vips no pueden entrar.

La valoración de esta cuestión será del siguiente modo:

- **(1 punto)** Modificación del programa principal para la creación de los 3 tipos de hilos según el contenido del fichero de entrada, y la modificación de variables globales y recursos de sincronización necesarios.
- **(1 punto)** Modificación de las condiciones de entrada de los clientes normales y vip.
- **(2 puntos)** Creación o modificación de la función de entrada para incluir a los clientes special.
- **(1 punto)** Modificación de la función de salida de la discoteca.

Se proporciona un fichero de entrada de prueba (ejemplo.txt), una posible salida del programa sería:

<pre> \$./disco ejemplo.txt client 0 is normal client 1 is normal client 2 is special client 3 is vip client 4 is special Starting client 3 Client 3 (vip) entering the disco Client 3 (vip) dancing in disco client 5 is normal client 6 is vip Starting client 5 Client 5 (normal) entering the disco Client 5 (normal) dancing in disco Starting client 0 Client 0 (normal) entering the disco Client 0 (normal) dancing in disco Starting client 4 Client 4 (special) waiting on the queue Starting client 2 Client 2 (special) waiting on the queue Starting client 1 Client 1 (normal) waiting on the queue Starting client 6 Client 6 (vip) waiting on the queue Client 0 (normal) exits the disco Client 1 (normal) waiting on the queue Client 2 (special) waiting on the queue Client 4 (special) waiting on the queue Client 6 (vip) waiting on the queue Client 5 (normal) exits the disco Client 3 (vip) exits the disco Client 1 (normal) waiting on the queue Client 2 (special) waiting on the queue Client 6 (vip) waiting on the queue Client 4 (special) entering the disco Client 4 (special) dancing in disco Client 2 (special) entering the disco Client 2 (special) dancing in disco Client 6 (vip) waiting on the queue Client 1 (normal) waiting on the queue </pre>	<pre> Client 4 (special) exits the disco Client 1 (normal) waiting on the queue Client 6 (vip) waiting on the queue Client 2 (special) exits the disco Client 1 (normal) waiting on the queue Client 6 (vip) entering the disco Client 6 (vip) dancing in disco Client 1 (normal) entering the disco Client 1 (normal) dancing in disco Client 6 (vip) exits the disco Client 1 (normal) exits the disco </pre>
---	--