CENTRO UNIVERSITARIO
DE TECNOLOGÍA Y ARTE DIGITAL

GRADO EN INGENIERÍA SOFTWARE
SISTEMAS OPERATIVOS, 2020-2021

PROYECTO I

Este proyecto vale un 15% de la nota final de la asignatura (el proyecto II valdrá un 20%). Es necesario obtener un 4 para poder liberar esta materia.

En esta práctica se pide desarrollar un simulador de memoria caché de un sistema ficticio de los años 70 llamado SUPERTRONIX,

SUPERTRONIX tenía un bus de memoria de 10 bits y usaba memoria real, con una caché de 4 líneas con correspondencia directa y 8 bytes por línea. Con lo que hemos aprendido en teoría, sabemos que la caché interpretará cada dirección de memoria recibida de la CPU en tres campos: palabra (3 bits), línea (2 bits) y etiqueta (5 bits).

Hay que desarrollar en C sobre *Linux* un proceso CACHEsym. El proceso dispone de un array de 4 elementos, del tipo `T_LINEA_CACHE`, cuya definición es:

```
typedef struct {
    short int ETQ;
    short int Datos[8];
} T_LINEA_CACHE;
```

El proceso tendrá una variable `tiempoglobal` que inicializará a valor 0. También creará otra con el nombre `numfallos` inicializada a 0.

En el arranque, CACHEsym inicializa a `FF` (hexadecimal) los campos `ETQ` y a `0` todos los campos de datos de la caché. Luego lee el fichero binario `RAM.bin` en la variable `RAM`, que es un array de 1024 `unsigned char`. A continuación, comienza la lectura del fichero de texto `accesos_memoria.txt` que contiene una lista de direcciones de memoria en hexadecimal, una por línea. El proceso tiene que tratar adecuadamente los errores si alguno de los ficheros no existe y avisar con el mensaje correspondiente. En ese caso terminará el proceso con `exit(-1)`.

Se repetirá el siguiente protocolo:

- CACHEsym lee una dirección del fichero `accesos_memoria.txt`.

- Obtiene el número de línea y comprueba si la etiqueta de la dirección es igual a `ETQ` de la línea de la caché.
- Si no es así, incrementa el valor de `numfallos` y escribe una línea con el texto "`T: %d, Fallo de CACHE %d, ADDR %04X ETQ %X linea %02X palabra %02X bloque %02X`", siendo T el instante. Se incrementa en 10 el contador `tiempoglobal`. Se copia el bloque correspondiente desde el array RAM y se imprime un mensaje indicando que se está cargando el bloque X en la línea Y. Se actualizan tanto el campo `ETQ` como los 8 bytes de datos de la línea.
- Por pantalla se escribe "`T: %d, Acierto de CACHE, ADDR %04X ETQ %X linea %02X palabra %02X DATO %02X`". Cada carácter leído se añade a una variable llamada texto, que es un array de 100 caracteres como máximo (no hace falta usar memoria dinámica).
- El proceso vuelca el contenido de la caché por pantalla con el siguiente formato:

```
T: 82, Fallo de CACHE 8, ADDR 0010 ETQ 0 linea 02 palabra 00 bloque 02
Cargando el bloque 02 en la linea 02
T: 92, Acierto de CACHE, ADDR 0010 ETQ 0 linea 02 palabra 00 DATO 71
ETQ:10  Datos 74 73 72 71 70 6F 6E 6D
ETQ:18  Datos 69 68 67 66 65 64 63 62
ETQ:0   Datos 78 77 76 75 74 73 72 71
ETQ:0   Datos 67 66 65 64 63 62 61 79
```

  Los datos se imprimen de izquierda a derecha de mayor a menor peso. Esto significa que el byte situado más a la izquierda es el byte 7 del bloque y el situado a la derecha el byte 0.

- EL proceso hace un `sleep()` de 2 segundos.

Al final se imprimirá un mensaje con el número total de accesos, número de fallos y tiempo medio de acceso.

Debajo, otro mensaje con el texto leído carácter a carácter desde la caché.

Entrega de la práctica

Cada equipo desarrollará el proyecto en `github`. Los equipos pueden ser de 2 ó 3 alumnos (excepcionalmente). Solo hay que subir una versión por equipo pero tiene que haber commits de todos ellos. Subirá un fichero `README.txt` con los nombres de los estudiantes, el fuente `CACHEsym.c` y el volcado de la salida de la ejecución en un fichero llamado `cache_log.txt`. Para ello ejecutar:

```
./CACHEsym > cache_log.txt
```

Evaluación

- El programa arranca correctamente, inicializa las variables, lee el fichero `RAM.bin` y abre el `accesos_memoria.txt` con el debido control de errores (1,5 pt).
- El proceso escribe bien por pantalla el contenido de la caché (1 punto)

- El proceso calcula correctamente la línea de caché de una dirección dada y comprueba si la etiqueta que contiene es la misma (1,5 puntos).
- El proceso escribe correctamente en pantalla el contenido del byte seleccionado (1 punto).
- El proceso hace correctamente el sleep de 2 segundos (0,5 puntos).
- El proceso escribe correctamente el número de accesos y fallos, el tiempo medio y el texto leído (1 punto).
- El proceso termina correctamente al leer la última línea de `accesos_memoria.txt` (0,5 puntos).
- Los commits están correctamente documentados (1,5 puntos).
- Limpieza y documentación del código (1,5 puntos). Se usan constantes en lugar de valores numéricos, hay comentarios, se emplean máscaras de bits para extraer los datos de la memoria, se utilizan, al menos, 4 funciones.

La evaluación se hará usando ficheros `accesos_memoria.txt` y `RAM.bin` diferentes del que se proporciona para el desarrollo, por lo que aconsejamos probar con varias combinaciones de direcciones de entrada.