

# Sistemas Operativos II

## Práctica 1 – Llamadas al sistema y señales

### Núcleo del sistema Operativo

Universidad de Castilla-La Mancha

Javier Alonso Albusac Jiménez  
Escuela Superior de Informática :: Universidad de Castilla-La Mancha

El objetivo principal es poner en práctica los conocimientos adquiridos en el Tema 2: *Núcleo del Sistema Operativo*, con el manejo de **llamadas al sistema**, el **tratamiento de señales** y la **gestión de interrupciones**, además de automatizar ciertas tareas propias de la administración de sistemas operativos. En concreto, se pide al estudiante que haga uso de algunas de las siguientes funciones del estándar ANSI C para resolver el problema que se plantea: `exit`, `fork`, funciones para el manejo de ficheros, `wait`, `waitpid`, funciones pertenecientes a la familia `execv`, `kill`, `signal` y `pipe`.

Por un lado, el programa consta de un proceso principal *manager* encargado de crear y coordinar al resto de procesos. Dichos procesos deben acceder al fichero *estudiantes.txt*, el cual consta de un conjunto de líneas donde se especifica la siguiente información en cada una de ellas:

1. DNI del estudiante.
2. Modelo de examen que debe realizar.
3. Nota que obtuvo en el examen anterior.

El proceso *manager* debe crear tres tipos de procesos **PA**, **PB** y **PC**, cada uno de ellos con una labor diferente.

En primer lugar, el proceso *manager* debe crear un proceso **PA** y esperar a que finalice. El proceso **PA**, debe realizar una lectura del fichero *estudiantes.txt* para crear un directorio por cada estudiante. El nombre del directorio debe ser el mismo que el DNI del estudiante. Cuando este proceso finalice la creación de directorios debe notificárselo al proceso *manager*, con el objetivo de que pueda continuar con su trabajo.

Una vez creado un directorio para cada estudiante, el proceso *manager* debe crear un proceso **PB** y otro **PC** que deben trabajar de forma simultánea; el proceso *manager* debe esperar a la finalización de ambos. El proceso **PB** debe hacer una lectura del fichero *estudiantes.txt* para extraer el modelo de examen que debe realizar cada estudiante. Note como en el material aportado con la práctica, hay un directorio llamado *exámenes*, donde hay tres modelos de examen *A.pdf*, *B.pdf* y *C.pdf*. El proceso **PB** debe copiar en cada directorio de estudiante el modelo que le corresponda. El modelo que le corresponde a cada estudiante se indica en el fichero *estudiantes.txt*, en concreto, en el segundo campo o valor de cada línea.

Por otro lado, un estudiante supera la prueba si la media aritmética del resultado obtenido en el examen anterior, junto con el que va a obtener con la realización de alguno de los modelos de examen, llega a la nota de corte de 5 (ambas pruebas tienen el mismo peso). La nota obtenida en el último examen para cada estudiante también está registrada en el tercer campo del fichero *estudiantes.txt*. El proceso *manager* debe crear un proceso **PC**, que trabaje de forma paralela al proceso **PB**. El proceso **PC** debe acceder al fichero *estudiantes.txt* y determinar, por cada estudiante, la nota que debe obtener en la nueva prueba para que la media finalmente alcance la nota de corte. El proceso **PC** debe crear un fichero de texto con extensión `.txt` que contenga la siguiente cadena: *«La nota que debes obtener en este nuevo examen para superar la prueba es X.»*, siendo *X*

la nota calculada. El fichero de texto debe estar almacenado en el directorio del estudiante. En resumen, cada directorio de estudiante debe contener un modelo de examen y un archivo de texto que le informe de la nota que debe obtener en el segundo examen, para que la media junto con la primera nota obtenida alcance la nota de corte.

Además, el proceso *manager* debe mantener un archivo *log.txt* donde se registre la finalización de cada una de las tareas y la media de las notas obtenidas por los estudiantes en la primera prueba. Dicha media debe ser calculada por **PC** y enviada al proceso *manager* a través de una tubería o *pipe*. Un ejemplo del contenido que debe tener el archivo *log.txt* puede ser el siguiente.

\*\*\*\*\* Log del sistema \*\*\*\*\*

Creación de directorios finalizada.

Copia de modelos de examen, finalizada.

Creación de archivos con nota necesaria para alcanzar la nota de corte, finalizada.

La nota media de la clase es: 7,25

FIN DE PROGRAMA

### Finalización del programa:

El programa finaliza si se producen alguna de las dos siguientes situaciones:

1. Los procesos **PB** y **PC**, finalizan de forma correcta.
2. El usuario pulsa CTRL+C durante la ejecución de alguno de los procesos, para interrumpir la ejecución de forma voluntaria. En tal caso, el proceso *manager* debe crear un cuarto proceso **PD** que se encargue de limpiar todo el contenido creado, es decir, debe eliminar todos los directorios y archivos asociados a los estudiantes. Antes de eliminar los contenidos, el proceso *manager* debe forzar la finalización de los procesos creados para evitar que continúen realizando cambios. Por último, se debe registrar en *log.txt* que se ha producido una interrupción voluntaria ordenada por el propio usuario.

Por último, la práctica debe incluir un proceso que trabaje en **segundo plano** (demonio) y que se active **periódicamente** cada minuto. El proceso debe realizar una copia de seguridad con todos los archivos relativos a la práctica en un directorio aparte. La finalización de este proceso puede realizarse de forma manual mediante el uso de órdenes en el intérprete de comandos.