

PRÁCTICA 4:

Llamadas al sistema relacionadas con procesos (I).

Objetivo:

Implementación de programas en C bajo UNIX, que utilicen las siguientes llamadas al sistema relacionadas con procesos:

<i>fork</i>	<i>execl</i>	<i>execv</i>
<i>getpid</i>	<i>exit</i>	<i>wait</i>

Enunciados:

Grupo A

Realizar un programa que funcione como un SHELL, mostrando el PROMPT del sistema y leyendo de la entrada estándar el nombre del comando a ejecutar, los parámetros y el entorno. El programa lanza un proceso hijo que ejecuta el comando correspondiente y queda a la espera de su terminación, mostrando la condición de salida del hijo antes de terminar.

Grupo B

Realizar un programa al que le pasamos desde la línea de comandos las rutas de dos imágenes de carga con sus parámetros respectivos separadas una de otra por el carácter “=”. El programa debe lanzar dos procesos hijo, el primer hijo ejecuta la primera imagen de carga, el segundo ejecuta la siguiente imagen de carga. El primer hijo redirige su salida estándar al fichero “sal_h1”, mientras que el segundo la redirige a “sal_h2”. El padre espera la terminación de los dos hijos, mostrando un mensaje cada vez que uno termina.

Ej `ejecparal ls -l / = ps -e`

Grupo C

Realizar un programa al que se le pase como primer parámetro en la línea de comandos el nombre de un directorio. El programa debe recorrer el directorio, mostrando para cada una de sus entradas el nombre de ruta absoluto y su tamaño en la salida estándar. Cuando la entrada corresponda a un directorio, el programa debe lanzar un proceso hijo que ejecute la misma imagen de carga y recorra dicho directorio.

Grupo D

Realizar un programa al que se le pasen tres parámetros en la línea de comandos, correspondientes a tres nombres de directorios. El programa debe lanzar un hijo para cada directorio que recorra todas sus entradas y muestre su nombre de ruta absoluto en la salida estándar. Cada proceso hijo debe redirigir su salida estándar a un fichero del directorio por defecto cuyo nombre será “salida1” para el primer directorio especificado, “salida2” para el segundo y “salida3” para el tercero.

Grupo E

Realizar un programa al que se le pase como primer parámetro en la línea de comandos el nombre de un fichero de texto que contiene en cada línea un comando seguido de sus argumentos (similar a un shell script). El programa leerá cada línea del fichero, lanzará un hijo por cada una de ellas que se encargue de ejecutar el comando correspondiente, esperará por su terminación e informará de su condición de salida.

Grupo F

Realizar un programa que tome como parámetro el nombre de un fichero de texto que contiene en cada línea un nombre de directorio. El programa debe lanzar un proceso hijo que recorrerá recursivamente los directorios según su orden de especificación en el fichero, mostrando por pantalla la ruta completa de cada una de sus entradas y su tamaño en bytes.

Grupo G

Realizar un programa al que se le pase como primer parámetro en la línea de comandos el nombre de un fichero de texto que contiene en cada línea un comando seguido de sus argumentos (similar a un shell script). El programa leerá cada línea del fichero y lanzará un hijo por cada una de ellas que se encargue de ejecutar el comando correspondiente, esperando por su terminación. El programa debe informar del identificador de proceso del padre y de todos los hijos ejecutados.

Grupo H

Realizar un programa que tome como parámetro el nombre de un fichero de texto que contiene en cada línea un nombre de directorio. El programa debe lanzar un proceso hijo (demonio) que periódicamente lea dicho fichero de texto recorriendo cada uno de los directorios especificados en el fichero y escribiendo los nombres de sus entradas en un fichero de texto llamado “salida”. Para la simulación, consideramos que el proceso hijo se ejecuta solamente 4 veces, teniendo que transcurrir un periodo de 10 segundos entre dos ejecuciones consecutivas.

Grupo I

Realizar un programa que tome como parámetro el nombre de un fichero de texto que contiene en cada línea un nombre de directorio. El programa debe lanzar un proceso hijo para cada uno de los directorios especificados que recorra sus entradas, y muestre la siguiente información en la salida estándar:

nombre_entrada tamaño identificación_usuario

Cada proceso hijo debe redireccionar su salida estándar a un fichero del directorio por defecto cuyo nombre se forma añadiendo el prefijo “salida” al lugar ocupado por el directorio en el fichero de texto (por ejemplo, la salida correspondiente al primer directorio especificado se redireccionará al fichero “salida1”).

Grupo J

Realizar un programa al que se le pasen como parámetros en la línea de comandos el nombre de dos ficheros. El programa debe lanzar un hijo que terminará con `exit(0)` si el primer fichero especificado en la línea de comandos está vacío o bien con `exit(1)` si el fichero no está vacío. El proceso padre examinará el retorno del hijo y escribirá en la salida estándar “fichero vacío” o bien “fichero no vacío” en función del resultado devuelto por el hijo. El proceso padre debe redireccionar su salida estándar al fichero especificado como segundo parámetro en la línea de comandos.