Ejercicios Introducción ASO

Usar las funciones disponibles en el API del sistema en los siguientes casos. En cada ejercicio añadir los ficheros de cabecera (#include) necesarios para que no se produzcan avisos en la compilación.

Ejercicio 1. Añadir el código necesario para gestionar correctamente los errores generados por la llamada setuid(). Usando la página de manual comprobar el propósito de la llamada setuid y su prototipo.

```
int main()
{
	/* Comprobar la ocurrencia de error y notificarlo con la llamada adecuada */
	setuid(0);
	return 1;
}
```

Ejercicio 2. En el código anterior imprimir el código de error generado por la llamada, tanto en su versión numérica como la cadena asociada.

Ejercicio 3. Escribir un programa que recorra en un bucle todos los mensajes de error disponibles en el sistema y los imprima.

Ejercicio 4. El comando del sistema uname(1) muestra información sobre diversos aspectos del sistema. Escribir un programa que muestre, claramente identificado, cada uno de los detalles del sistema que reporta uname. Consultar uname(2) para más información sobre la llamada al sistema.

Ejercicio 5. La función sysconf(3) permite consultar información sobre la configuración del sistema. Escribir un programa que muestre los Ticks por segundo, el número máximo de procesos hijos y el número máximo de ficheros.

Ejercicio 6. Escribir un programa que obtenga la información de configuración del sistema y del sistema de ficheros llamando a sysconf(3) y pathconf(3). El programa deberá mostrar:

- El número máximo de procesos simultáneos que puede ejecutar un usuario.
- El tamaño de las páginas de memoria
- La longitud máxima de los argumentos a un programa
- El número máximo de ficheros que puede abrir un proceso
- El número máximo de enlaces de un fichero
- El tamaño máximo de una ruta
- El tamaño máximo de un nombre de fichero

Ejercicio7. Escribir un programa que muestre los uid efectivos y real de un usuario y que indique si el correspondiente ejecutable tiene activado el bit setuid. Mejorar a continuación el programa anterior para que se muestre además el nombre de login, el directorio home e información de usuario del usuario real.

Ejercicio8. La función principal para obtener la hora del sistema es time(). Escribir un programa que obtenga la hora usando está función y la muestre en el terminal. Mejorar a continuación el programa para que se muestre además la hora en formato legible, usando la función ctime. ¿Dónde se reserva espacio para el valor de la cadena que devuelve la función? ¿Es necesario liberar el puntero?

Ejercicio 9. Cuando es necesario obtener la información horaria con precisión de microsegundos se puede usar gettimeofday(). Escribir un programa que mida cuánto tarda un bucle de 10000 repeticiones en incrementar una variable en una unidad en cada iteración.