



1. Hacer un programa que abra archivos con flags `O_EXCL` y `O_CREAT`. Verificar el comportamiento esperado cuando ambos flags existen y el archivo existe previamente.
2. Hacer un programa que utilice `umask` y luego `open` para crear varios archivos con distintos permisos por default.
3. Hacer un programa que liste los archivos en el directorio actual y su tamaño. En caso que existan subdirectorios, listar también sus contenidos de la misma forma. Utilizar `opendir/readdir/closedir`.
4. Hacer un programa que genere un file con agujeros grandes utilizando `lseek`. Recuperar la información del file con `fstat` e imprimir los campos de la estructura `stat`. Comparar con lo que muestran los comandos `ls` y `du` sobre el archivo. [agujeros.txt tiene 5072 bytes de las 3 formas](#)
5. Diseñe un programa que escriba archivos que contengan 10 bloques de 4 bytes cada uno separados entre ellos por “huecos” de 32, 512, 8KB y 1MB (cada archivo con un tamaño de huecos distinto). Recuperar la información de los archivos con `fstat` y analizar los campos `st_size` y `st_blocks`. Explique sus conclusiones.  
6. Usando las funciones `link` y `symlink` verificar como se afecta el reference count de un ínode. Verificar con la función `stat(2)`. Hacer lo mismo con los comandos “`ln ...`” y “`ln -s ...`” verificando con el comando `stat`.
7. Hacer un programa que obtenga memoria en un loop mientras pueda con `new` y otro que lo haga con `mmap` anónimo. Cuánta memoria se puede obtener con cada método de allocación? Ver los mapas de memoria de los procesos antes que terminen.