

## Ejercicio

En un sistema UNIX hay un archivo denominado datos cuyas características obtenidas con el mandato `ls -ls datos` son las siguientes:

```
nodo-i   bloques   protección  dueño grupo   bytes archivo
87       8           rw-r----- pepe alumnos  7642 datos
```

Responda a las preguntas siguientes:

- Si `juan` es del mismo grupo que `pepe`, ¿Puede el alumno `juan` escribir el archivo? ¿Puede leerlo? ¿Qué debería hacer `pepe` para que `juan` pudiera ejecutarlo?
- ¿Qué ocurriría en UNIX si un usuario tuviera permisos de acceso `r--` a un archivo y su grupo tuviera `rw-`? ¿Podría el usuario escribir el archivo?
- Implemente un programa que cambie los permisos de acceso a un archivo UNIX, de forma similar al comando `chmod`.

## Solución

- Dado que los permisos del grupo son `r--`, los miembros del mismo grupo que `pepe` solo pueden leer sus archivos. Luego `juan` no puede escribirlo y sí leerlo. Para poderlo ejecutar, los permisos deberían ser `r-x`, es decir `101`, es decir `5` en octal. Luego se debería ejecutar el comando:

```
>> chmod 0650
```

- Si un usuario tuviera accesos `r` a un archivo y su grupo tuviera `rw` el usuario NO podría leer el archivo, dado que con las Access Control List se pueden dar permisos o quitar permisos a un usuario específico.
- Se puede implementar un programa de forma similar a `chmod` con el código que se muestra continuación, que da todos los permisos para el archivo `"/home/hola.txt"`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/stat.h>

int main(int argc, char **argv)
{
    int mode = 0777;
    char file[100] = "/home/hola.txt";

    if (chmod (file,mode) < 0)
    {
        fprintf (stderr, "%s: error en chmod(%s, %o) - %d (%s)\n",
                argv[0], file, mode, errno, strerror(errno));
    }
}
```



```
    exit(1);  
}  
return(0);  
}
```