**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULDAD DE INGENIERÍA**

**SISTEMAS OPERATIVOS**

*PROGRAMA: “ADMINISTRACIÓN DE ARCHIVOS Y PLANIFICACIÓN DE DISCO”*

PROFESORA: *LAURA SANDOVAL MONTAÑO*

GRUPO: 06

INTEGRANTES:

* *CHÁVEZ DELGADO JORGE LUIS*
* *VARGAS CASTRO DANIEL*

**SEMESTRE 2017-1**

1° de diciembre de 2016

**OBJETIVO**

El presente programa tiene el fin de mostrar el seguimiento de una simulación de la administración de archivos en un disco, así como su planificación. Se aplicarán los conocimientos vistos en clase como son la técnica de asignación de espacio por encadenamiento de bloques y se aplicará la técnica de planificación de disco de “primero el de menor tiempo de búsqueda”.

Con el programa realizado y el siguiente documento, comprobaremos que entendimos los conceptos vistos en clase y somos capaces de aplicarlos en una situación real.

**DESCRIPCIÓN DE LA SOLUCIÓN**

**Administrador de archivos:**

Para representar los archivos, declaramos una estructura con los campos “nombre” que contendrá el nombre del archivo y “pista” que almacenará la pista donde se encuentra cada bloque del archivo en la memoria.

La simulación del disco se hizo mediante un arreglo de estructuras del tipo mencionado anteriormente. Cada elemento del arreglo representa un bloque del disco, de este modo, podremos utilizar los índices del arreglo para mandar llamar directamente, sin búsquedas, al bloque que sea necesario y acceder fácilmente a los datos que contenga. Además, dada la naturaleza de los arreglos, ya hay un símil con el encadenamiento de bloques, técnica con la que se deben administrar los archivos en el disco para este ejercicio.

Una vez declarados los elementos mencionados anteriormente y otras variables auxiliares (como contadores, apuntadores a archivo, banderas, etc), se hizo una lectura del archivo con los nombres y tamaños de los archivos ficticios, almacenándolos en el elemento (bloque) del arreglo (disco) que le corresponda de acuerdo a su tamaño. Por ejemplo, si el archivo “tarea.docx” pesa 3kB, se reservarán 3 casillas del arreglo para almacenar dicho archivo, y por la naturaleza de las estructuras que utilizamos, en esas 3 casillas estará almacenado el nombre del archivo y la pista que le corresponda a cada una de esas casillas (recordemos que cada casilla representa un bloque y el arreglo completo representa el disco). Adicionalmente, conforme se van insertando elementos al arreglo, se irán imprimiendo sus datos, mostrando así qué archivos ocupan cuáles bloques.

Tras analizar detalladamente los requerimientos, notamos que la planificación del disco, para este caso particular, se iba a comportar como una especie de pila, de modo que el último archivo en cargarse al disco será el primero en ser leído (LIFO). Fue por eso que durante la lectura del archivo con archivos ficticios se fue almacenando, en orden inverso, el bloque donde inicia cada uno de los archivos ficticios. Por ejemplo, si “tarea.docx” empieza en el bloque 1, “song.mp3” en el bloque 5 y “prog.c” en el 20, en un nuevo archivo se almacenarán únicamente los números 20, 5 y 1 en ese orden, y realmente no hay que preocuparse por cuál archivo estará más cerca de la cabeza, dado que ya no llegarán más solicitudes y el último archivo colocado en el disco siempre será el más cercano a la cabeza.

Finalmente, y antes de iniciar la planificación, se almacena en una variable el índice del último elemento del arreglo al que se accedió, o sea el bloque, simulando así la posición inicial de la cabeza lectora y se asigna una bandera que indica a qué dirección apunta la cabeza. Estos datos cambiarán constantemente en la parte de planificación.

**Planificador de disco:**

Para lograr la planificación, empezamos con la lectura del archivo que contiene los bloques de inicio de cada archivo ficticio, el cual se crea automáticamente durante la parte de administración.

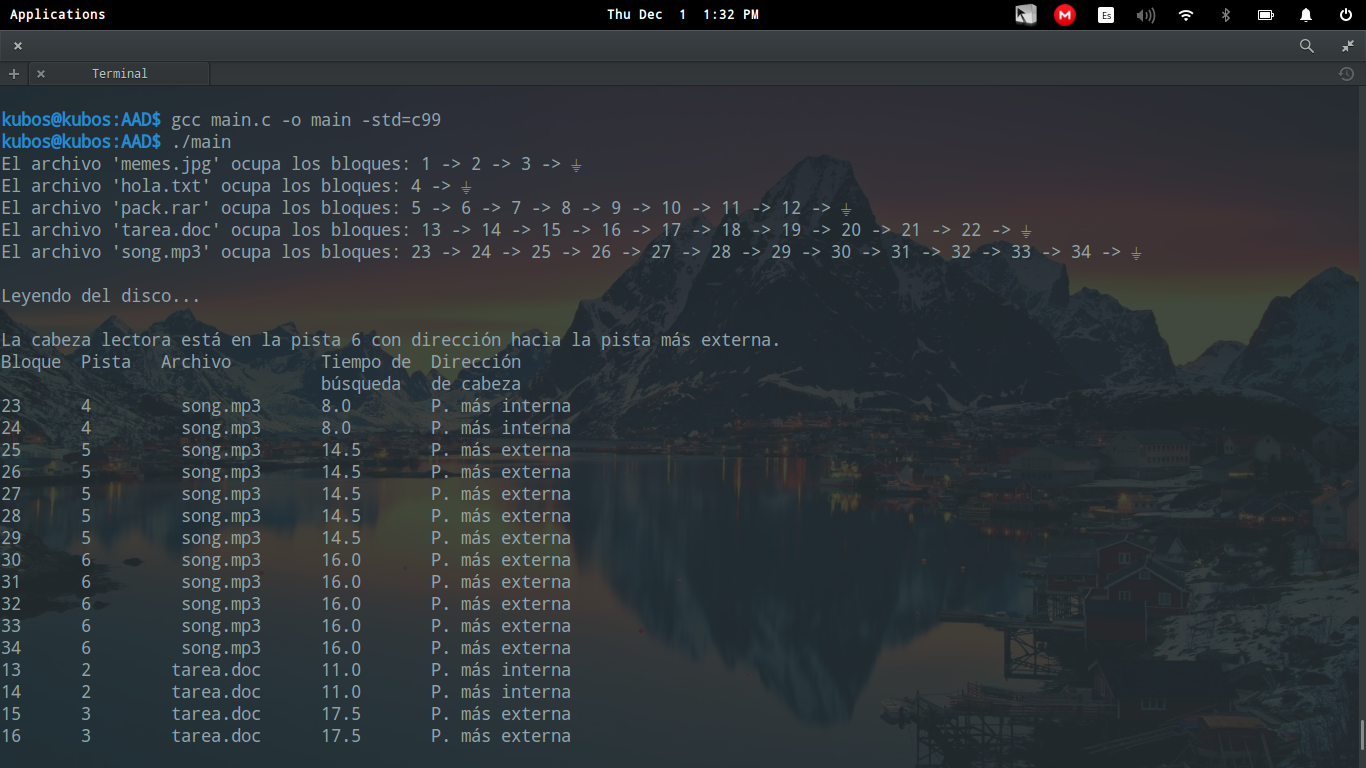
Para cada “bloque” en el archivo, se accede directamente a ese bloque del disco utilizando el dato leído como índice del arreglo. Una vez ubicado el archivo en ese bloque, se recorre el disco hasta que se encuentra el último bloque que ocupa dicho archivo. Conforme se recorre el arreglo (o sea, el disco) se va imprimiendo, en forma de tabla, el nombre, la pista en que está el bloque actual (recordemos que esos datos están almacenados en cada elemento del arreglo) y el tiempo que le está tomando leer ese archivo en específico, considerando los saltos de pista y cambios de dirección de la cabeza, dato que también es mostrado en la tabla como referencia para el usuario.

Finalmente, se imprime el tiempo ficticio total que le tomó al programa leer los archivos con ayuda de una variable que acumula los tiempos de búsqueda de cada archivo.

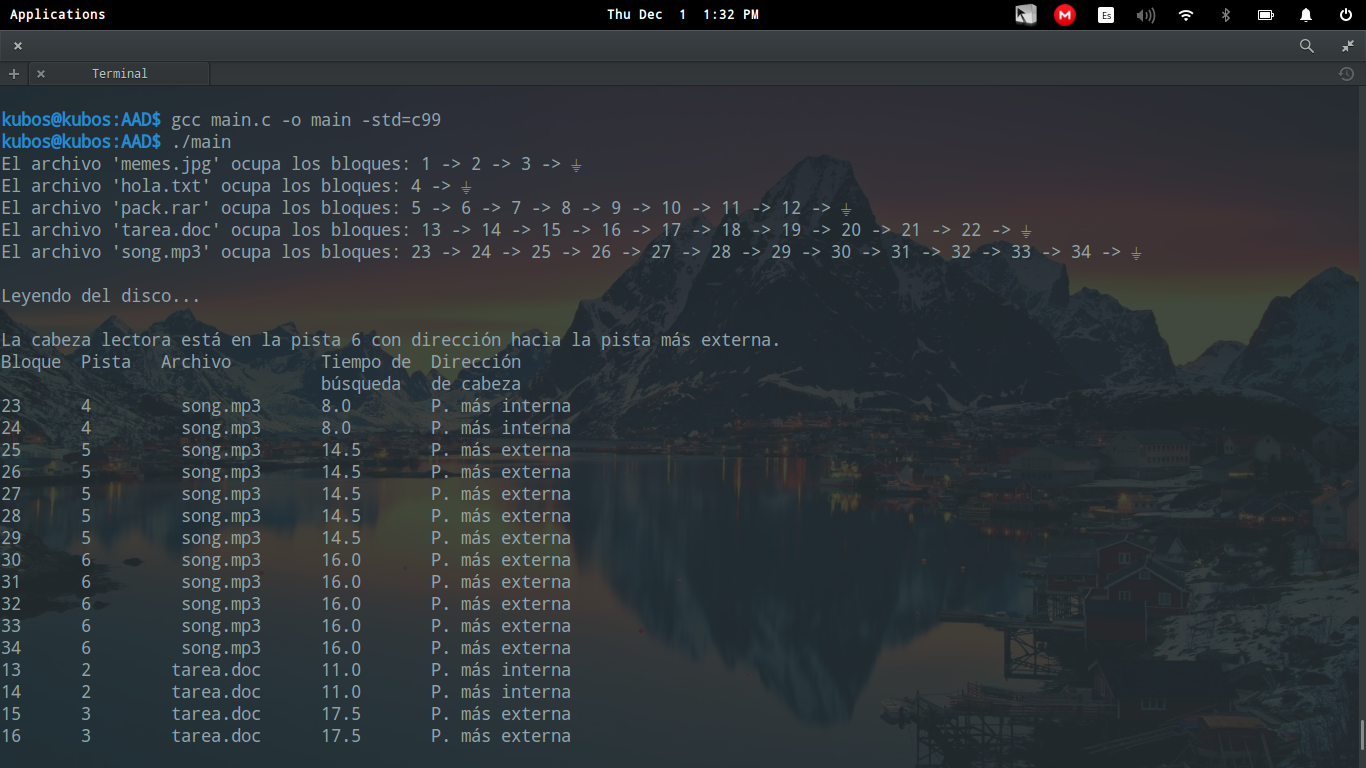
**¿CÓMO PUEDO CREAR EL EJECUTABLE Y EJECUTAR EL PROGRAMA?**

Para crear el ejecutable, desde nuestra terminal nos movemos a la ubicación del archivo “main.c”. Utilizaremos el compilador GCC, ejecutando los comandos de la siguiente forma:

***$ gcc main.c -o main -std=c99***

******

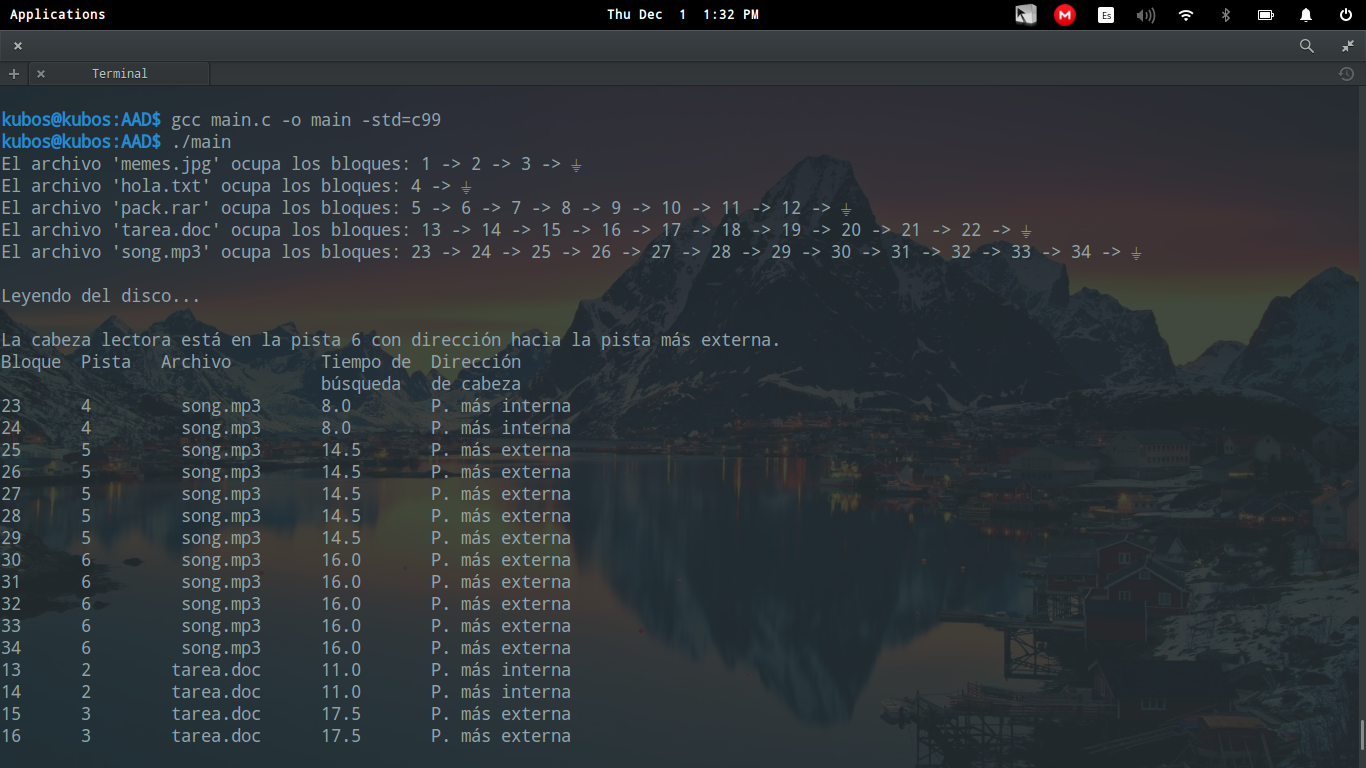
Lo que hará este comando es generarnos un ejecutable llamado “***main****”.* La bandera ***-std=c99*** es para que no mande ***warnings*** a la hora de compilar debido a que utilizamos caracteres especiales en las impresiones en panatalla. Solo selecciona la version del compilador, no afecta al funcionamiento del programa.

**

*Para correrlo, ejecutamos el comando:*

***$ ./main***

Y tendremos en ejecución el programa:



**CONCLUSIONES**

de Chávez Delgado Jorle Luis:

Durante la realización del programa pudimos darnos cuenta de la utilidad de la administración y planificación de cualquier sistema, en este caso fue el de un disco, el objetivo se cumplió satisfactoriamente. Además se pudo observar claramente que para poder realizar un programa de este tipo es primordial entender la teoría vista en clase así como los ejercicios realizados en la misma.

Finalmente es impresionante poder ver la simulación de lo que realmente hace el disco dentro de nuestras computadoras.

de Vargas Castro Daniel:

La administración y planificación de la lectura de los archivos en un disco es fundamental para el correcto funcionamiento de cualquier computadora e incluso smartphones hoy en día, por lo que como alumno de ingeniería en computación este proyecto me pareció ideal para que entendamos, aunque sea básicamente, cómo funciona un disco duro, pues si bien nuestro programa no está hecho en un lenguaje de nivel tan bajo como el que se usa realmente en un disco, estamos probando que entendemos la idea y en un momento dado podemos aplicar el conocimiento en un problema real de ingeniería.

Sin embargo, lo que más me llevo de este proyecto, es que, a pesar de que la técnica de planificación estaba bien definida en teoría, ésta se puede comportar de forma diferente una vez llevado a la práctica, lo cual podría en un momento dado mejorar o empeorar la eficiencia del disco, pero logra su objetivo independientemente de ello.