

Laboratorio de Computación

II

Archivos

Archivos

- Conjunto de bytes que son almacenados en un dispositivo.
 - Los registros deben ser de longitud fija.
 - Los registros debieran ser identificados con un valor único e irrepetible.
 - El archivo, en consecuencia al ítem anterior, no debiera admitir registros duplicados.

Archivos

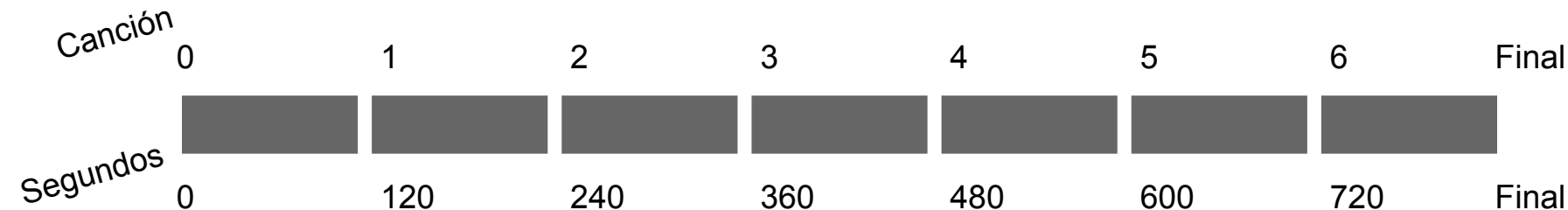
- Imaginemos que tenemos un cassette de audio. En él, cada canción dura exactamente dos minutos y se encuentran una inmediatamente después de la otra.

No nos es posible ir fácilmente de la canción 2 a la canción 5. Pero sabemos que la canción 1 comienza en el segundo 0. La canción 2 en el segundo 120. Por lo tanto, la canción 5 en el segundo 480.



Aclaración para centennials: La imagen de arriba corresponde a un cassette de audio.

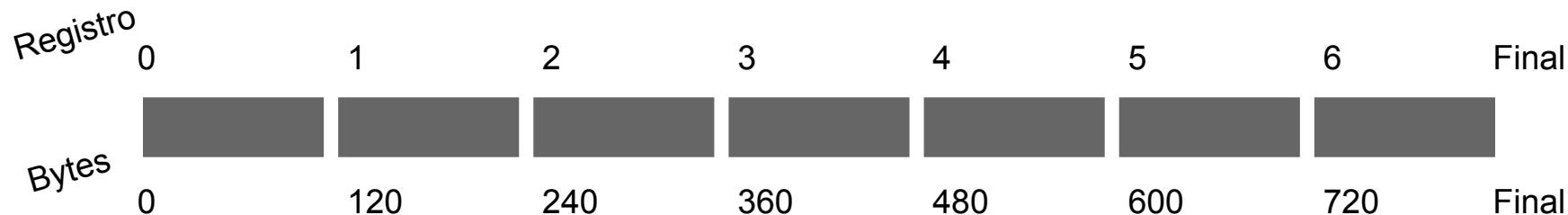
Archivos



- Si quisiera ir del inicio de la canción 0 al inicio de la canción 3. No puedo decirle "Adelantar 3 canciones desde la canción 0" pero sí puedo decirle "Adelantar 360 segundos desde el segundo 0".
- Si estoy ubicado en la canción 3 y quisiera ir a la canción 1 podría "Retroceder 240 segundos desde el segundo 360".

Recuerden: 1 canción  = 120 segundos

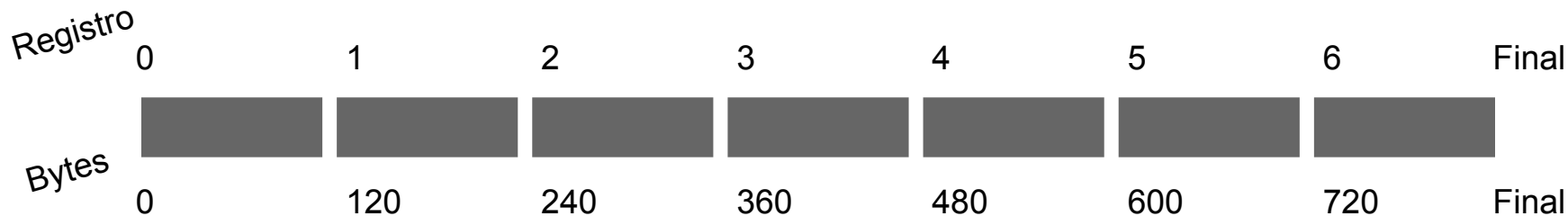
Archivos



- Pensemos la misma idea pero con archivos. Reemplacemos Canción por Registro y Segundos por Bytes. En lugar de música tenemos bits.
- Ya entendieron cómo se almacena información en un archivo binario.

Recuerden: 1 registro  = 120 bytes

Archivos



- La función que nos permite adelantar o retroceder un archivo se llama `fseek` y recibe los siguientes parámetros.

```
int fseek(FILE *archivo, long int desplazamiento, int origen);
```

fseek

```
int fseek(FILE *archivo, long int desplazamiento, int origen);
```

archivo - Es el puntero file al archivo que queremos manipular. Debe estar abierto previamente.

desplazamiento - Es la cantidad de bytes que queremos desplazarnos.

origen - Es una bandera que indica desde donde queremos desplazarnos la cantidad de bytes indicada por *desplazamiento*.



Valor	Alias	Descripción
0	SEEK_SET	Inicio del archivo
1	SEEK_CUR	Posición actual del archivo
2	SEEK_END	Final del archivo

fseek - Ejemplos

```
int fseek(FILE *archivo, long int desplazamiento, int origen);
```

<code>fseek(p, 500, SEEK_SET);</code>	Se desplaza 500 bytes desde el inicio del archivo p
<code>fseek(p, 500, SEEK_CUR);</code>	Se desplaza 500 bytes desde la posición actual del archivo p
<code>fseek(p, 0, SEEK_END);</code>	Se desplaza 0 bytes desde el final del archivo p
<code>fseek(p, 4 * sizeof(XX), 0);</code>	Se desplaza 4 veces el tamaño del tipo XX desde el inicio del archivo p
<code>fseek(p, pos * sizeof(XX), 0);</code>	Se desplaza pos veces el tamaño del tipo XX desde el inicio del archivo p

ftell

```
long int ftell(FILE *archivo);
```

Devuelve la cantidad de bytes desde el inicio del archivo a la posición donde se encuentre al momento de ejecutar la función.

```
fseek(p, 0, SEEK_END);
```



¿Qué tiene la variable t?

```
t = ftell(p);
```

```
cr = t / sizeof(XX);
```



¿Qué tiene la variable cr?

Ejemplo en C/C++