



Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica
malloc()

Explicación de la práctica 5

Manejo de archivos

Seminario de Lenguajes opción C

Facultad de Informática
Universidad Nacional de La Plata

2022



Indice

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica

malloc()

① Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

② Memoria dinámica

malloc()



Formas de lectura-escritura de archivos

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- Se pueden leer y escribir archivos:

- Por bloques → fread/fwrite.
- Por carácter → fgetc/fputc.
- Por línea → fgets/fputs.



Formas de lectura-escritura de archivos

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- Se pueden leer y escribir archivos:
 - Por bloques → fread/fwrite.
 - Por carácter → fgetc/fputc.
 - Por línea → fgets/fputs.



Formas de lectura-escritura de archivos

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- Se pueden leer y escribir archivos:
 - Por bloques → fread/fwrite.
 - Por carácter → fgetc/fputc.
 - Por línea → fgets/fputs.



Formas de lectura-escritura de archivos

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- Se pueden leer y escribir archivos:
 - Por bloques → fread/fwrite.
 - Por carácter → fgetc/fputc.
 - Por línea → fgets/fputs.



Por bloques (fread/fwrite)

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- Es práctica para archivos binarios.
- Pero se puede usar archivos de texto.
- Se indican la cantidad de “elementos” a leer y el tamaño de cada “elemento”.
- Las funciones retornan la cantidad de “elementos” leídos.



Por bloques (fread/fwrite)

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- Es práctica para archivos binarios.
- Pero se puede usar archivos de texto.
- Se indican la cantidad de “elementos” a leer y el tamaño de cada “elemento”.
- Las funciones retornan la cantidad de “elementos” leídos.



Por bloques (fread/fwrite)

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- Es práctica para archivos binarios.
- Pero se puede usar archivos de texto.
- Se indican la cantidad de “elementos” a leer y el tamaño de cada “elemento”.
- Las funciones retornan la cantidad de “elementos” leídos.



Por bloques (fread/fwrite)

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- Es práctica para archivos binarios.
- Pero se puede usar archivos de texto.
- Se indican la cantidad de “elementos” a leer y el tamaño de cada “elemento”.
- Las funciones retornan la cantidad de “elementos” leídos.



Lectura de un archivo binario

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

```
FILE *arch = fopen("nombre_de_archivo", "rb");
int leidos;
long nro;
leidos = fread(&nro, sizeof(long), 1, arch);
while (leidos > 0){
    printf("Se leyo -> %ld\n", nro);
    leidos = fread(&nro, sizeof(long), 1, arch);
}
fclose(arch);
```



Lectura de un archivo binario

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica
malloc()

```
FILE *arch = fopen("nombre_de_archivo", "rb");
int leidos;
long nro;
leidos = fread(&nro, sizeof(long), 1, arch);
while (leidos > 0){
    printf("Se leyo -> %ld\n", nro);
    leidos = fread(&nro, sizeof(long), 1, arch);
}
fclose(arch);
```

Este código no es portable



Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Problemas en una aplicación real:

- No hay garantías sobre el tamaño de long.
 - `stdint.h` → `int32_t`, `uint32_t`, `int64_t`, etc...¹.
 - `inttypes.h` → `PRId32`, `PRlu32`, `SCNd32`, etc...².
- ¿Big-endian o little-endian?³.
 - `netinet.in`
 - `htons`, `htonl` → Big endian al orden de la máquina.
 - `ntohs`, `ntohl` → Orden de la máquina a big endian.

¹ https://www.gnu.org/software/libc/manual/html_node/Integers.html

² <http://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/inttypes.5.html>

³ <http://www.ibm.com/developerworks/aix/library/au-endianc/index.html>



Archivos de texto

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica
malloc()

- Por carácter:

```
FILE *arch = fopen("archivo.txt", "r");  
int c;  
while ((c = fgetc(arch)) != EOF && c != '\n'){  
    putchar(c);  
}
```

- Por línea:

```
char linea[1024]; // Suponemos...  
if (fgets(linea, 1024, arch) != NULL){  
    printf("Resultado: %s\n", linea);  
}
```

- No podemos calcular la longitud de una línea hasta leerla.
- ¿Cómo la almacenamos?
- Suponemos y ajustamos...



Archivos de texto

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- Por carácter:

```
FILE *arch = fopen("archivo.txt", "r");  
int c;  
while ((c = fgetc(arch)) != EOF && c != '\n'){  
    putchar(c);  
}
```

- Por línea:

```
char linea[1024]; // Suponemos...  
if (fgets(linea, 1024, arch) != NULL){  
    printf("Resultado: %s\n", linea);  
}
```

- No podemos calcular la longitud de una línea hasta leerla.
- ¿Cómo la almacenamos?
- Suponemos y ajustamos...



Archivos de texto

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica
malloc()

- Por carácter:

```
FILE *arch = fopen("archivo.txt", "r");  
int c;  
while ((c = fgetc(arch)) != EOF && c != '\n'){  
    putchar(c);  
}
```

- Por línea:

```
char linea[1024]; // Suponemos...  
if (fgets(linea, 1024, arch) != NULL){  
    printf("Resultado: %s\n", linea);  
}
```

- No podemos calcular la longitud de una línea hasta leerla.
- ¿Cómo la almacenamos?
- Suponemos y ajustamos...



Archivos de texto

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- Por carácter:

```
FILE *arch = fopen("archivo.txt", "r");  
int c;  
while ((c = fgetc(arch)) != EOF && c != '\n'){  
    putchar(c);  
}
```

- Por línea:

```
char linea[1024]; // Suponemos...  
if (fgets(linea, 1024, arch) != NULL){  
    printf("Resultado: %s\n", linea);  
}
```

- No podemos calcular la longitud de una línea hasta leerla.
- ¿Cómo la almacenamos?
- Suponemos y ajustamos...



Archivos de texto

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica
malloc()

- Por carácter:

```
FILE *arch = fopen("archivo.txt", "r");
int c;
while ((c = fgetc(arch)) != EOF && c != '\n'){
    putchar(c);
}
```

- Por línea:

```
char linea[1024]; // Suponemos...
if (fgets(linea, 1024, arch) != NULL){
    printf("Resultado: %s\n", linea);
}
```

- No podemos calcular la longitud de una línea hasta leerla.
- ¿Cómo la almacenamos?
- Suponemos y ajustamos...



Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica

malloc()

Repaso de memoria dinámica y después volvemos a archivos...



Memoria dinámica - repaso

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

`malloc()`

- **`malloc()`**: Aloca un bloque de memoria.
- **`calloc()`**: Igual que `malloc()` pero lo inicializa en 0.
- **`realloc()`**: Cambia el tamaño de un bloque alocado con alguna de las anteriores.
- **`free()`**: Libera la memoria alocada con los anteriores.



Memoria dinámica - repaso

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

`malloc()`

- **`malloc()`**: Aloca un bloque de memoria.
- **`calloc()`**: Igual que `malloc()` pero lo inicializa en 0.
- **`realloc()`**: Cambia el tamaño de un bloque alocado con alguna de las anteriores.
- **`free()`**: Libera la memoria alocada con los anteriores.



Memoria dinámica - repaso

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

`malloc()`

- **`malloc()`**: Aloca un bloque de memoria.
- **`calloc()`**: Igual que `malloc()` pero lo inicializa en 0.
- **`realloc()`**: Cambia el tamaño de un bloque alocado con alguna de las anteriores.
- **`free()`**: Libera la memoria alocada con los anteriores.



Memoria dinámica - repaso

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica

malloc()

- **malloc()**: Aloca un bloque de memoria.
- **calloc()**: Igual que `malloc()` pero lo inicializa en 0.
- **realloc()**: Cambia el tamaño de un bloque alocado con alguna de las anteriores.
- **free()**: Libera la memoria alocada con los anteriores.



malloc()

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

1 Una implementación típica de malloc():

- Mantiene una lista de bloques libres.
- Busca en la lista con criterio first fit (el primero que entre).
- Demanda memoria al Sistema Operativo.

2 free() marca como libre el bloque que corresponda.

3 realloc() aloca más espacio conservando los datos.



malloc()

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

1 Una implementación típica de malloc():

- Mantiene una lista de bloques libres.
- Busca en la lista con criterio first fit (el primero que entre).
- Demanda memoria al Sistema Operativo.

2 free() marca como libre el bloque que corresponda.

3 realloc() aloca más espacio conservando los datos.



malloc()

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

- ① Una implementación típica de malloc():
 - Mantiene una lista de bloques libres.
 - Busca en la lista con criterio first fit (el primero que entre).
 - Demanda memoria al Sistema Operativo.
- ② free() marca como libre el bloque que corresponda.
- ③ realloc() aloca más espacio conservando los datos.



malloc()

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

- ① Una implementación típica de malloc():
 - Mantiene una lista de bloques libres.
 - Busca en la lista con criterio first fit (el primero que entre).
 - Demanda memoria al Sistema Operativo.
- ② free() marca como libre el bloque que corresponda.
- ③ realloc() aloca más espacio conservando los datos.



malloc()

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

- ① Una implementación típica de malloc():
 - Mantiene una lista de bloques libres.
 - Busca en la lista con criterio first fit (el primero que entre).
 - Demanda memoria al Sistema Operativo.
- ② free() marca como libre el bloque que corresponda.
- ③ realloc() aloca más espacio conservando los datos.



malloc()

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

- ① Una implementación típica de malloc():
 - Mantiene una lista de bloques libres.
 - Busca en la lista con criterio first fit (el primero que entre).
 - Demanda memoria al Sistema Operativo.
- ② free() marca como libre el bloque que corresponda.
- ③ realloc() aloca más espacio conservando los datos.



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```



■ Memoria allocada previamente

□ Memoria libre

■ Memoria allocada



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```



■ Memoria asignada previamente

□ Memoria libre

■ Memoria asignada



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```

Realloc



■ Memoria asignada previamente

□ Memoria libre

■ Memoria asignada



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

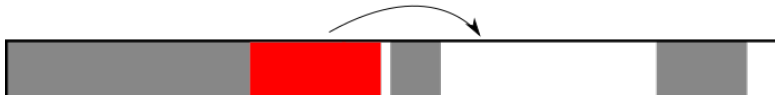
Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```

Copiar



■ Memoria asignada previamente

□ Memoria libre

■ Memoria asignada



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

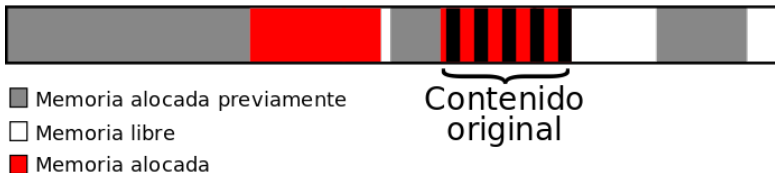
Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```





Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```



■ Memoria asignada previamente

□ Memoria libre

■ Memoria asignada



Ejemplo

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques




Lectura de un archivo binario

Memoria dinámica

malloc()

```
char *mem = malloc(sizeof(char) * 20);  
mem = realloc(mem, sizeof(char) * 30);  
mem = realloc(mem, sizeof(char) * 60);  
free(mem);
```



-  Memoria asignada previamente
-  Memoria libre
-  Memoria asignada



Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica

`malloc()`

Volviendo a archivos



fgets() y memoria dinámica

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

malloc()

- **fgets():** Lee la línea completa con el '\n' incluido.
- La última línea del archivo puede no tener '\n'.
- Si no hay '\n' también puede significar que no alcanza el espacio para seguir leyendo.



fgets() y memoria dinámica

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- **fgets():** Lee la línea completa con el '\n' incluido.
- La última línea del archivo puede no tener '\n'.
- Si no hay '\n' también puede significar que no alcanza el espacio para seguir leyendo.



fgets() y memoria dinámica

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- **fgets():** Lee la línea completa con el '\n' incluido.
- La última línea del archivo puede no tener '\n'.
- Si no hay '\n' también puede significar que no alcanza el espacio para seguir leyendo.



Ejemplo (sin comprobar)

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques
Lectura de un
archivo binario

Memoria
dinámica

malloc()

```
char *leer_linea(FILE *f){
    char *mem = malloc(1024 * sizeof(char));
    char *ptr = mem;
    while (fgets(ptr, 1024, f) != NULL){
        if (strlen(ptr) < 1023 || ptr[1022] == '\n')
        {
            // Se leyó la línea completa
            return mem;
        }
        mem = realloc(mem, (strlen(mem) + 1024) *
            sizeof(char));
        ptr = mem + strlen(mem);
    }
    if (ferror(f)){
        free(ptr);
        return NULL;
    }
    return mem;
}
```




getline()

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- En GLibC (la biblioteca C de GNU) existe getline().
- Aloca memoria por nosotros.
- Siempre lee la línea completa
- No es estándar.
- Pero si usamos GCC + GLibC podemos usarla.

```
#define _GNU_SOURCE
#include <stdio.h>
/* ... */
char *mem = NULL;
size_t tam = 0;
size_t leidos = getline(&mem, &tam, arch);
if (leidos < 0) puts("Error o fin de archivo");
/* ... */
free(mem);
```



getline()

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- En GLibC (la biblioteca C de GNU) existe getline().
- Aloca memoria por nosotros.
- Siempre lee la línea completa
- No es estándar.
- Pero si usamos GCC + GLibC podemos usarla.

```
#define _GNU_SOURCE
#include <stdio.h>
/* ... */
char *mem = NULL;
size_t tam = 0;
size_t leidos = getline(&mem, &tam, arch);
if (leidos < 0) puts("Error o fin de archivo");
/* ... */
free(mem);
```



getline()

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- En Glibc (la biblioteca C de GNU) existe getline().
- Aloca memoria por nosotros.
- Siempre lee la línea completa
- No es estándar.
- Pero si usamos GCC + Glibc podemos usarla.

```
#define _GNU_SOURCE
#include <stdio.h>
/* ... */
char *mem = NULL;
size_t tam = 0;
size_t leidos = getline(&mem, &tam, arch);
if (leidos < 0) puts("Error o fin de archivo");
/* ... */
free(mem);
```



getline()

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- En GLibC (la biblioteca C de GNU) existe getline().
- Aloca memoria por nosotros.
- Siempre lee la línea completa
- No es estándar.
- Pero si usamos GCC + GLibC podemos usarla.

```
#define _GNU_SOURCE
#include <stdio.h>
/* ... */
char *mem = NULL;
size_t tam = 0;
size_t leidos = getline(&mem, &tam, arch);
if (leidos < 0) puts("Error o fin de archivo");
/* ... */
free(mem);
```



getline()

Explicación de
la práctica 5

Seminario de
Lenguajes
opción C

Introducción

Formas de
lectura-escritura de
archivos

Por bloques

Lectura de un
archivo binario

Memoria
dinámica

malloc()

- En Glibc (la biblioteca C de GNU) existe getline().
- Aloca memoria por nosotros.
- Siempre lee la línea completa
- No es estándar.
- Pero si usamos GCC + Glibc podemos usarla.

```
#define _GNU_SOURCE
#include <stdio.h>
/* ... */
char *mem = NULL;
size_t tam = 0;
size_t leidos = getline(&mem, &tam, arch);
if (leidos < 0) puts("Error o fin de archivo");
/* ... */
free(mem);
```



Ejemplos en clase

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques
Lectura de un archivo binario

Memoria dinámica

`malloc()`

- Leer enteros con `strtoul`.
- Leer fechas con formato: `dd/mm/yyyy`.
- Leer varias líneas y combinarlas.



Las manpages son tu mejor aliado

Explicación de la práctica 5

Seminario de Lenguajes opción C

Introducción

Formas de lectura-escritura de archivos

Por bloques

Lectura de un archivo binario

Memoria dinámica

malloc()

- man fread
- man fgets
- man sscanf
- man malloc
- man getline
- man ...
- <http://man.cx>

Otra documentación útil:

- <http://www.gnu.org/software/libc/manual/>
- <http://www.gnu.org/software/gnu-c-manual/>