CC3301 Programación de software de sistemas

Archivos

Archivos de texto

- Necesario: #include <stdio.h>
- Tipo de datos: FILE
- Ej.: FILE *in, *out;
- Abrir un archivo:
  - Lectura: FILE *in= fopen("pers.txt", "r");
  - Escritura: FILE *out= fopen("datos.txt", "w");
- Leer texto:

  int fgetc(FILE *stream);

  char *fgets(char *s, int size, FILE *stream);

Lee un byte. Entrega EOF si se acabó el archivo

Lee una línea en un string de a lo más size-1 bytes. Se garantiza \0, pero \n va solo si hay espacio.

- Escribir texto:

  int fputc(c, FILE *stream);

  int fputs(const char *s, FILE *stream);

Ejemplo: mostrar contenido de un archivo de texto (fgets.c)

```c
#include <stdio.h>

int main() {
  char buf[82];
  FILE *in= fopen("pers.txt", "r");
  for (;;) {
    if (fgets(buf, 82, in)==NULL)
      break;
    printf("%s", buf);
  }
  return 0;
}
```

Ejemplo: reemplazar \n por un espacio en blanco (fputs.c)

```c
#include <stdio.h>
#include <string.h>

int main() {
  char buf[82];
  FILE *in= fopen("pers.txt", "r");
  FILE *out= fopen("datos.txt", "w");
  for (;;) {
    if (fgets(buf, 82, in)==NULL)
      break;
    int len= strlen(buf);
    buf[len-1]= ' '; // Cambia '\n' por ' '
    fputs(buf, out);
  }
  return 0;
}
```

Archivos

- Con formato:

  int fprintf(FILE *stream, const char *format, ...);

  int fscanf(FILE *stream, const char *format, ...);

También se puede leer con formato

- Otras funciones:

  int feof(FILE *stream); *// chequea fin de arch.*

  int ferror(FILE *stream); *// chequea error*

  int fseek(FILE *stream, long offset, int whence);

  // Acceso directo a archivos

  int fclose(FILE *stream); *// cierra archivo*

- Entrada estándar, salida estándar y salida estándar de errores:

Ya vienen declarados.

  FILE *stdin, *stdout, *stderr;

- Funciones para acceder a entrada/salida estándar:

  getc, gets, putc, puts, printf, scanf

X

No llevan *stream*

Ejemplo: escribir un archivo de texto (pers-txt.c)

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct {
  char rut[12];
  int edad;
  char nom[30];
}  Persona;

Persona pers[] = { { "14475267-1", 20, "pedro" },
                   { "14847282-4", 22, "juan" },
                   { "14844738-9", 24, "diego" },
};

int main() {
  FILE *out= fopen("pers.txt", "w");
  for (int i= 0; i<3; i++) {
    fprintf(out, "%s,%d,%s\n", pers[i].rut, pers[i].edad, pers[i].nom);
  }
  fclose(out); // Importante!

  return 0;
}
```

```
$ make pers-txt
cc -g -Wall -pedantic -std=c99  -g  pers-txt.c   -o pers-txt
pss@debian11mate:~/CC3301/Slides/Archivos
$ ./pers-txt
pss@debian11mate:~/CC3301/Slides/Archivos
$ cat pers.txt
14475267-1,20,pedro
14847282-4,22,juan
14844738-9,24,diego
```

Archivos binarios

- Lectura/escritura de archivos binarios (o texto)

  size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);

  size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);

- Ejemplo: escribir y leer arreglo de personas

```
typedef struct {
  char rut[12];            // char *pers;
  int edad;
  char nom[30];            // char *nom;
}  Persona;

Persona pers[30];

FILE *out, *in;
...
fwrite(pers, sizeof(Persona), 30, out);
fclose(out); // Importante!
...
FILE *in= fopen("pers.txt", "r");
Persona *pers2= malloc(100*sizeof(Persona));
int cnt= fread(pers2, sizeof(Persona), 100, in);
// cnt==30
```

- Problema: si se escribe en una máquina little endian y se lee en una big endian, la edad va a ser basura

Ejemplo: escribir y leer un archivo binario

// pers.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct {
  char rut[12];
  int edad;
  char nom[30];
}  Persona;

Persona pers[] = { { "14475267-1", 20, "pedro" },
                   { "14847282-4", 22, "juan" },
                   { "14844738-9", 24, "diego" },
};

int main() {
  FILE *out= fopen("pers.bin", "w");
  fwrite(pers, sizeof(Persona), 3, out);
  fclose(out); // Importante!

  FILE *in= fopen("pers.bin", "r");
  Persona *pers2= malloc(100*sizeof(Persona));
  int cnt= fread(pers2, sizeof(Persona), 100, in);

  for (int i= 0; i<cnt; i++) {
    printf("%s %d %s\n", pers2[i].rut, pers2[i].edad, pers2[i].nom);
  }

  return 0;
}
```

```
$ make ./pers
cc -g -Wall -pedantic -std=c99  -g  pers.c   -o pers
pss@debian11mate:~/CC3301/Slides/Archivos
$ ./pers
14475267-1 20 pedro
14847282-4 22 juan
14844738-9 24 diego
```

```
14475267-1^@^@^T^@^@^@pedro^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^
@14847282-4^@^@^V^@^@^@juan^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^
@^@14844738-9^@^@^X^@^@^@diego^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@
```

Ejemplo incorrecto (pers2.c)

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct {
  char *rut;
  int edad;
  char *nom;
}  Persona;

Persona pers[] = { { "14475267-1", 20, "pedro" },
                   { "14847282-4", 22, "juan" },
                   { "14844738-9", 24, "diego" },
};

int main() {
  FILE *out= fopen("pers.bin", "w");
  fwrite(pers, sizeof(Persona), 3, out);
  fclose(out); // Importante!

  FILE *in= fopen("pers.bin", "r");
  Persona *pers2= malloc(100*sizeof(Persona));
  int cnt= fread(pers2, sizeof(Persona), 100, in);

  for (int i= 0; i<cnt; i++) {
    printf("%s %d %s\n", pers2[i].rut, pers2[i].edad, pers2[i].nom);
  }

  return 0;
}
```

```
$ make pers2
cc -g -Wall -pedantic -std=c99  -g  pers2.c   -o pers2
pss@debian11mate:~/CC3301/Slides/Archivos
$ ./pers2
14475267-1 20 pedro
14847282-4 22 juan
14844738-9 24 diego
```

Ejemplo: mostrar pers.bin (pers3.c)

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct {
  char *rut;
  int edad;
  char *nom;
}  Persona;

int main() {
  FILE *in= fopen("pers.bin", "r");
  Persona *pers2= malloc(100*sizeof(Persona));
  int cnt= fread(pers2, sizeof(Persona), 100, in);

  for (int i= 0; i<cnt; i++) {
    printf("%s %d %s\n", pers2[i].rut, pers2[i].edad, pers2[i].nom);
  }

  return 0;
}
```

```
$ make -B pers3
cc -g -Wall -pedantic -std=c99  -g  pers3.c   -o pers3
pss@debian11mate:~/CC3301/Slides/Archivos
$ ./pers3
Violación de segmento
```

¡Nunca escriba punteros en un archivo binario!