Sistemas operativos embebidos

Manejo de Archivos en C

Manejadores de Archivo

- Antes de poder usar un archivo es necesario declarar un apuntador a una estructura de tipo FILE que servirá para hacer referencia a dicho archivo

  ```
  #include <stdio.h>

  FILE *archivo;
  ```

- Archivos declarados por stdio.h
  - stdin entrada de consola
  - stdout salida de consola
  - Stderr salida para mensajes de error

Tipos de archivos

- Texto
  - Codificación UTF-8, MAC-ROMAN, WIN
  - Acceso secuencial
- Binarios
  - Acceso aleatorio
  - Usualmente se almacenan estructuras del mismo tipo y se manejan de modo similar a un arreglo en memoria
- Todos los archivos deben cerrarse con

  int fclose(FILE *fp);

Apertura de un archivo

- FILE *fopen(const char *path, const char *mode);
- Abre el archivo cuyo nombre se encuentra en la cadena apuntada por path
- Mode:
  - r solo lectura, posicionado al inicio del archivo
  - r+ lectura y escritura, posicionado en el inicio
  - w el archivo es creado o truncado a tamaño 0 para escritura. Posicionado en el inicio
  - w+ similar a w, pero también puede leerse
  - a abierto para agregar (apending). Se crea el archivo si no existe. Posicionado al final del archivo
  - a+ similar a a, pero también puede leerse

Manejo de errores al abrir un archivo

```
arch=fopen(“archivo”,”r”);
if(arch == NULL) { // Manejo del error
    printf(“Hubo un error al abrir el archivo”);
    exit(0);
}
fclose(arch);
```

Funciones para manejo de archivos de texto

- int fprintf(FILE *stream, const char *format, ...);
- int fgetc(FILE *stream);
- char *fgets(char *s, int size, FILE *stream);
- int fscanf(FILE *stream, const char *format, ...);
- int sscanf(const char *str, const char *format, ...);
- int feof(FILE *stream);

Ejemplo de lectura renglón por renglón de un archivo de texto

```
FILE *entrada;
char cadena[100];
entrada=fopen("entrada.txt","r");
if(entrada==NULL){
    printf("Error abriendo el archivo de entrada");
    return(-1);
}
while(fgets(cadena,98,entrada)!=NULL)
    printf(“%s”,cadena);
fclose(entrada);
```

Funciones para manejo de archivos binarios

- 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);
- int fseek(FILE *stream, long offset, int whence);
- long ftell(FILE *stream);
- void rewind(FILE *stream);
- int fgetpos(FILE *stream, fpos_t *pos);
- int fsetpos(FILE *stream, fpos_t *pos);

Ejemplo archivos binarios

```c
#include <stdio.h>

int main(int argc, char **argv)
{
    FILE *salida,*entrada;
    int variable=0x3061;

    salida=fopen("salida.bin","w");
    fwrite((void *) &variable, sizeof(int),1,salida);
    fclose(salida);

    entrada=fopen("salida.bin","r");
    fread((void *) &variable, sizeof(int),1,entrada);
    fclose(entrada);
    printf("%x",variable);
    return 0;
}
```

Clases para manejo de archivos en C++

- Archivo de entrada: ifstream
- Archivo de salida: ofstream
- Archivo para entrada y salida: fstream
- Métodos para manejar archivos
  - Abrir un archivo: open()
  - Escribir y leer a un archivo (texto): << y >>
  - Escribir y leer a un archivo binario: write() y read()
  - Cerrar un archivo: close()

Ejemplo archivo de texto

```
#include <iostream>
#include <fstream>
using namespace std;

int main () {
  ofstream myfile;
  myfile.open ("example.txt");
  myfile << "Writing this to a file.\n";
  myfile.close();
  return 0;
}
```