Archivos

Lectura con Char

Métodos de Acceso

- Usa el método ifstream y get para leer el archivo char por char
- Usar la función getc para leer el archivo char por char
- Usar la función fgetc para leer el archivo Char por Char

Método ifstream y get para leer el archivo char por char

La forma más común de tratar con la E/S de archivos a la manera de C++ es usar std::ifstream. Al principio, un objeto ifstream se inicializa con el argumento del nombre del archivo que necesita ser abierto. Fíjate que la declaración if verifica si la apertura de un archivo tuvo éxito. Luego, usamos la función incorporada get para recuperar el texto dentro del archivo char por char y push_back en el contenedor vector. Finalmente, damos salida a los elementos vectoriales a la consola para propósitos de demostración.

Método 1

```
#include <iostream>
#include <fstream>
#include <vector>

using std::cout; using std::cerr;
using std::endl; using std::string;
using std::ifstream; using std::vector;
```

Método 1

```
int main(){
    string filename("input.txt");
    vector<char> bytes;
    char byte = 0;
     ifstream input_file(filename);
    if (!input_file.is_open()) {
        cerr << "Could not open the file - '" << filename << "'" << endl;
        return EXIT_FAILURE;  }
```

Método 1

```
while (input_file.get(byte)) { bytes.push_back(byte);}
    for (const auto &i : bytes) {
        cout << i << "-";
    }
    cout << endl;
    input_file.close();
    return EXIT_SUCCESS;
}
```

Función getc para leer el archivo char por char

Otro método para leer el archivo carácter por carácter es usar la función getc, que toma el flujo de FILE * como argumento y lee el siguiente carácter si está disponible. Luego, el flujo de archivos de entrada debe ser iterado hasta que el último char sea alcanzado, y eso se implementa con la función feof, comprobando si el final del archivo ha sido alcanzado. Tenga en cuenta que siempre se recomienda cerrar los flujos de archivos abiertos una vez que ya no se necesiten.

Método 2

```
#include <iostream>
#include <fstream>
#include <vector>

using std::cout; using std::cerr;
using std::endl; using std::string;
using std::ifstream; using std::vector;
```

Método 2

```
int main() {
    string filename("input.txt");
    vector<char> bytes;
    FILE* input_file = fopen(filename.c_str(), "r");
    if (input_file == nullptr) {  return EXIT_FAILURE; }
```

Método 2

```
unsigned char character = 0;
    while (!feof(input_file)) {
        character = getc(input_file);
        cout << character << "-";   }
    cout << endl;
    fclose(input_file);
    return EXIT_SUCCESS;
}
```

Función fgetc para leer el archivo Char por Char

fgetc es un método alternativo a la función anterior, que implementa exactamente la misma funcionalidad que getc. En este caso, el valor de retorno de fgetc se utiliza en la sentencia if ya que devuelve el EOF si se alcanza el final del flujo de archivos. Como se recomienda, antes de la salida del programa, cerramos el flujo con una llamada fclose.

Método 3

```
#include <iostream>

#include <fstream>

#include <vector>


using std::cout; using std::cerr;

using std::endl; using std::string;

using std::ifstream; using std::vector;
```

Método 3

```
int main(){
    string filename("input.txt");
    vector<char> bytes;
    FILE* input_file = fopen(filename.c_str(), "r");
    if (input_file == nullptr) {
       return EXIT_FAILURE;
    }
```

Método 3

```
int c;
    while ((c = fgetc(input_file)) != EOF) {
       putchar(c);
       cout << "-"; }
    cout << endl;
    fclose(input_file);
    return EXIT_SUCCESS;
}
```

Bibliografia

https://www.delftstack.com/es/howto/cpp/read-file-char-by-char-cpp/#usar-la-funci%C3%B3n-getc-para-leer-el-archivo-char-por-char