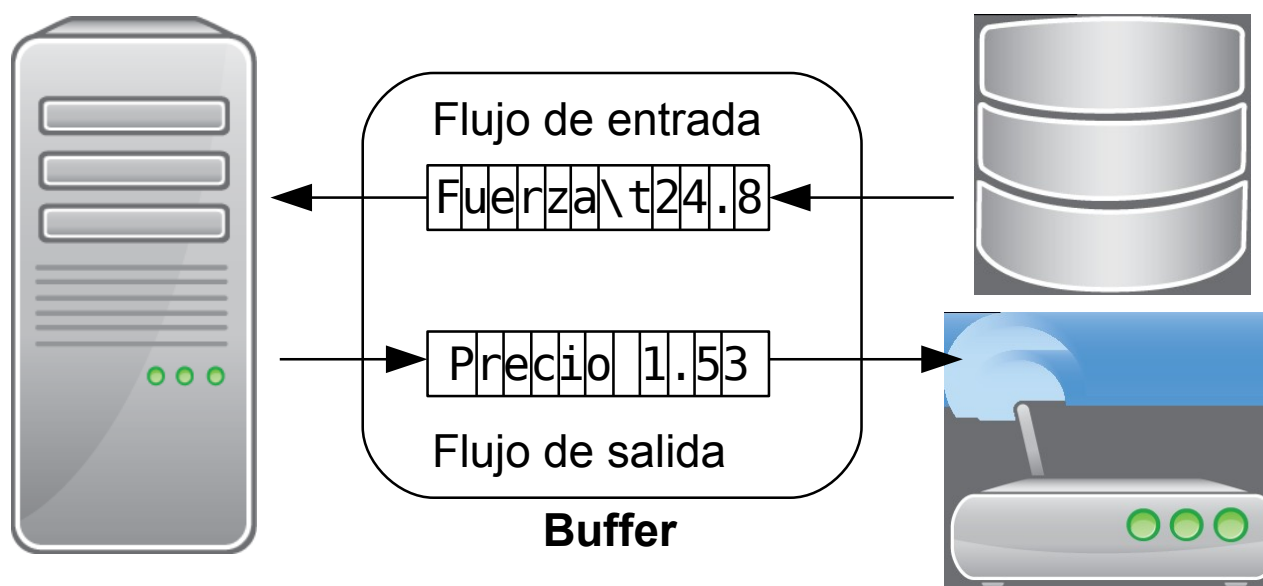




Flujos

- El intercambio de datos entre un programa (memoria RAM) y una fuente externa (Archivo, transmisión, etc) se modeliza como un flujo.
 - En nuestro caso será un flujo de caracteres, ya que nos orientaremos a “archivos de texto”





Flujos predefinidos

- Ya estuvimos trabajando con flujos: `cin` como flujo de entrada y `cout` como flujo de salida. Ambos definidos en `<iostream>`
- Características:
 - No hace falta definirlos
 - No hace falta abrirlos o cerrarlos
 - Se consideran “infinitos”
 - Se puede indicar fin de archivo con `ctrl+d` en linux y `ctrl+z` en windows (F6 ingresa `ctrl+z` también)
 - No siempre funciona bien, sobre todo en windows, después de `ctrl+z` suele necesitar un `enter` para que lo tome



Flujos con archivos

- Para establecer un flujo con un archivo es necesario incluir `<fstream>`
- Operatoria:
 - Definir el flujo
 - Abrirlo: vincularlo a un archivo
 - Operar: lectura y/o escritura (igual que con los flujos predefinidos)
 - Cerrarlo: desvincular del archivo (baja buffers si es necesario)



Flujo de solo escritura

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

using namespace std;

int main()
{
    ofstream archiesc; //ouput file stream: para escritura
    archiesc.open("Archivo.txt");
    if (!archiesc) {
        cout << "Error al abrir el archivo" << endl;
        return 1;
    }
    archiesc << "Primera línea" << endl;
    int dato = 53;
    archiesc << "Segunda línea con dato = " << dato << endl;
    archiesc.close()
    return 0;
}
```



Flujo de solo lectura

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

using namespace std;

int main()
{
    ifstream archilec; //input file stream: para lectura
    archilec.open("Archivo.txt");
    if (!archilec) {
        cout << "Error al abrir el archivo" << endl;
        return 1;
    }
    string str;
    archilec >> str;
    cout << str << endl; //muestra: Primera
    archilec >> str;
    cout << str << endl; //muestra: línea
    archilec.close();
    return 0;
}
```



Flujo de lectura y escritura

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

using namespace std;

int main()
{
    fstream archi;
    archi.open("Archivo.txt");
    //Falta control de éxito en abrir archivo
    archi << "Una frase cualquiera" << endl;
    string str;
    archi.seekg(0, ios::beg); //Me posiciona al principio
                             //del archivo

    archi >> str;
    cout << "Leído : " << str << endl; //Muestra: Una
    archi.close();
    return 0;
}
```



Posicionamiento

- Funciones para conocer la posición de los punteros de flujo (**g** por get: leer, y **p** por put: escribir)

```
streampos tellg(); //streampos es un alias de un entero
```

```
streampos tellp(); //sin signo del tamaño que haga falta
```

- Funciones para cambiar la posición de los punteros dentro del flujo

```
ostream& seekp(streampos pos); //respecto al inicio del archivo
```

```
ostream& seekp(streamoff off, ios_base::seekdir way);
```

- Es habitual que si el flujo es un archivo, se use un único puntero para ambos casos
- Posibles valores de seekdir:
 - beg: desde el inicio
 - cur: desde la posición actual
 - end: desde el final



Licencia

*Esta obra, © de Eduardo Zúñiga, está protegida legalmente bajo una licencia Creative Commons, **Atribución-CompartirDerivadasIgual 4.0 Internacional**.
<http://creativecommons.org/licenses/by-sa/4.0/>*

***Se permite: copiar, distribuir y comunicar públicamente la obra; hacer obras derivadas y hacer un uso comercial de la misma.
Siempre que se cite al autor y se herede la licencia.***

