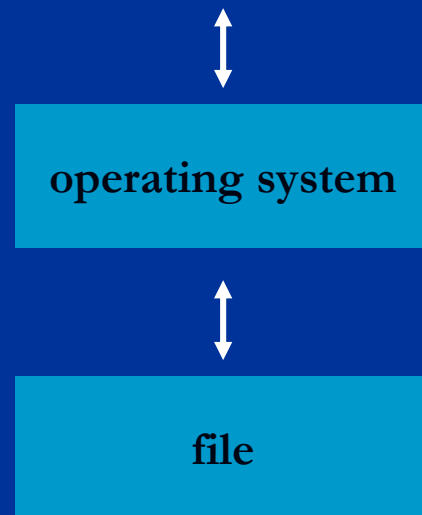
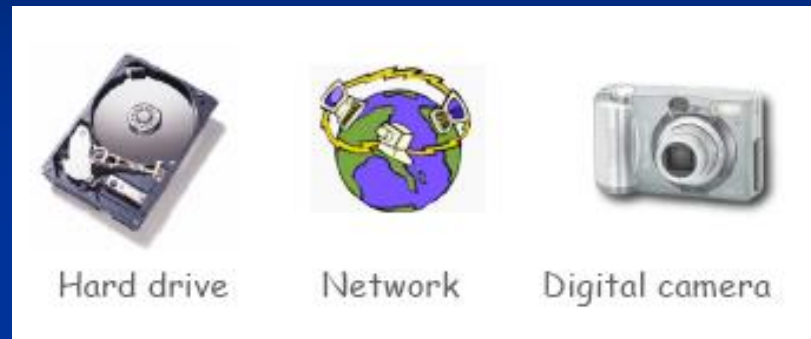


Files

Dr. Joshua Gluckman

Files



A file is an
abstraction for
communicating with
external devices

External files

- Location
 - “../temp/picture.jpg”
- Text files
 - Sequence of characters encoded in ascii
 - EOF character at end
 - Example: html
- Binary files
 - Sequence of bytes
 - Example: gif

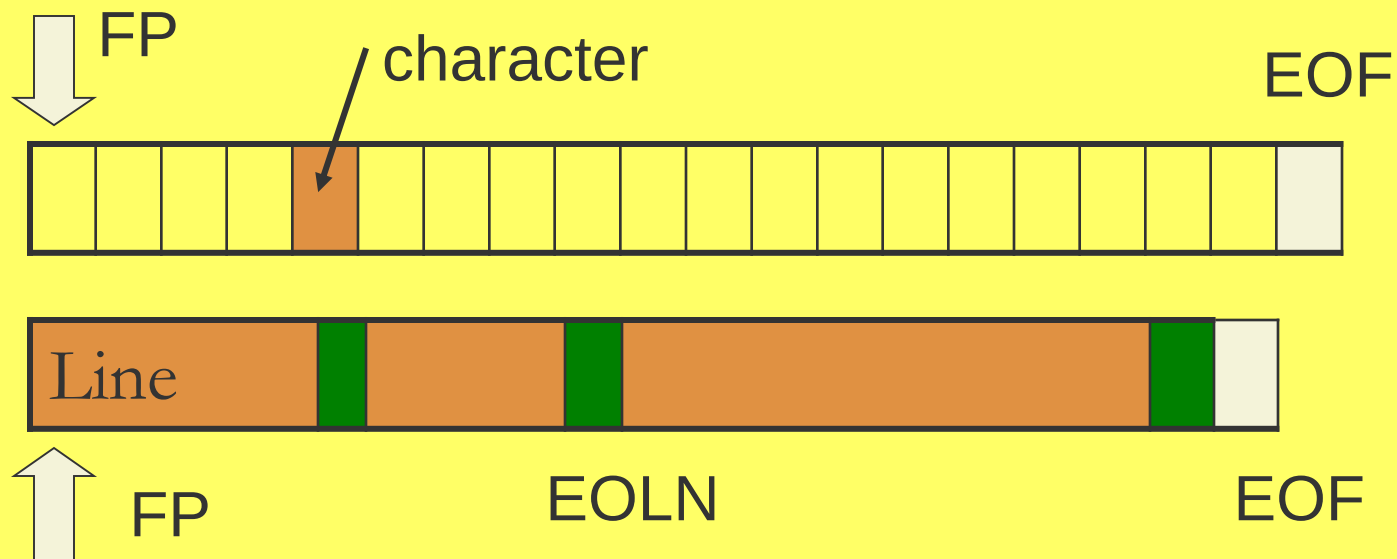
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html >
<head>
<title>The American University in Cairo (AUC)</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<link rel="shortcut icon" href="/favicon.ico" type="image/x-
icon" />

<style type="text/css">
<!--
```

html text file

Logical representation of text file

- Elements are single character or strings with EOLN
- Access is *Sequential*
- Current position given by FP (file pointer)



Streams

- C++ abstraction (object) for reading and writing to files
- Opening a stream
 - Read or write mode
 - Text or binary mode
- Reading / writing using text stream
 - Extraction and insertion operators << and >>
 - Stream member functions

Some stream member functions

- ifstream – input file stream object
 - open() and close()
 - fail() and eof()
 - get(), getline(), ignore(), read()
- ofstream – output file stream object
 - open() and close()
 - fail()
 - put() and write()

Example

- Write a program that makes a copy of a text file
- Open two streams, 1 input and 1 output
- Copy the text from one stream to the other

First try

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

void main (void) {

    string infile, outfile;
    ifstream source; ofstream target;
    string s;

    cout << "Enter Input File Name: ";    cin >> infile;
    cout << "Enter Output File Name: ";   cin >> outfile;

    source.open(infile.c_str()); target.open(outfile.c_str());

    while (! source.eof()) {
        source >> s;
        target << s;  }

    source.close();          target.close(); }
```

First try

This file only
contains a few
words.

Input file

Thisfileonlycontainsafewwords.

Copy

Second try

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

void main (void) {

    string infile, outfile;
    ifstream source; ofstream target;
    char c;

    cout << "Enter Input File Name: ";    cin >> infile;
    cout << "Enter Output File Name: ";    cin >> outfile;

    source.open(infile.c_str()); target.open(outfile.c_str());

    while (! source.eof()) {
        source.get(c);
        target.put(c);    }

    source.close();          target.close(); }
```

Second try

This file only
contains a few
words.

Input file

This file only
contains a few
words..

Copy

Third try

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

void main (void) {

    string infile, outfile;
    ifstream source; ofstream target;
    char c;

    cout << "Enter Input File Name: ";    cin >> infile;
    cout << "Enter Output File Name: ";    cin >> outfile;

    source.open(infile.c_str()); target.open(outfile.c_str());

    source.get(c);
    while (! source.eof()) {
        target.put(c);
        source.get(c);    }

    source.close();          target.close(); }
```

Example

- Write a program that counts the occurrence of each character in a text file

Example

- Write a program that counts the occurrence of each character in a text file

```
int count[128];

for (int i=0; i<128; i++) count[i]=0;

source.get(c);
while (! source.eof()) {
    source.get(c);
    int ascii = int(c);  \\ from character to ascii
    count[ascii]++; }

for (int i=0; i<128; i++)
    if (count[i]>0)      \\ from ascii to character
        cout << char(i) << ' ' << count[i] << endl;
```

Example

```
int count[128];

for (int i=0; i<128; i++) count[i]=0;

source.get(c);
while (! source.eof()) {
    source.get(c);
    int ascii = int(c);  \\ from character to ascii
    count[ascii]++; }

for (int i=0; i<128; i++)
    if (count[i]>0)      \\ from ascii to character
        cout << char(i) << ' ' << count[i] << endl;
```

Example: reading a .ppm image file



File peppers.ppm

The format of the PPM file looks like this:

```
P6  
width height max_value  
rgbrgbrgb...
```

```
P6  200 300 255 !2@...
```

Text

Binary data

Example: reading a .ppm image file

```
ifstream imageFile; string s, fileName;
int width, height, numLevels;

// ios::binary because it is a data file
imageFile.open(fileName.c_str(), ios::binary);

// read in the ppm header
imageFile >> s;
if (s != "P6") cout << "Not a valid PPM image" << endl;
imageFile >> width >> height >> numLevels;
imageFile.get(); // read in newline character

image = new char[width*height*3];

// read in the image data rgbbrgb ...
imageFile.read(image, width*height*3);
imageFile.close();
```