

FILE I/O PREREQUISITES

Operating system features impacting programs

Delroy A. Brinkerhoff

FUNDAMENTAL FILE SYSTEM FEATURES

- Directories (Folders) vs. Files
 - Files contain data and programs
 - Directories are specialized files
- Tree structure

Ę

- Root at the top: / or \
- Leaves at the bottom (no subtrees) are files
- Directories may or may not be logically empty



CURRENT WORKING DIRECTORY

- A program's host environment includes its current working directory (CWD)
 - Apparent location of a running program in the file system
 - . represents the CWD
 - ./program

- .\program
- Starting location for finding resources



ABSOLUTE FILE PATH NAME

- Always begins at the root (may include a drive letter)
- A unique path to a file or directory
- Independent of the program's CWD
- The same symbol naming the root is used for the file path separator
 - /B/D/G

Ę

- \B\D\G
- C:\B\D\G



RELATIVE FILE PATH NAME (I)

- Access or visibility relative to the program's CWD
- Represents the path's starting and ending
 - B/D/G

Ē

- B\D\G
- Inaccessible or invisible from A or C



RELATIVE FILE PATH NAME (2)

- Backtracking: going up before going down
- . . represents the parent or super directory
 - ../D/G
 - ..\D\G
 - • •
 - Can climb multiple levels
 - ../../x
 - ..\..\x



RELATIVE FILE PATH NAME (3)

- A relative path can consist of a single name
 - File
 - Directory
 - G



LINE SEPARATOR (TERMINATOR)

POSIX (E.G., LINUX)

See the quick\n red fox jump\n over the lazy\n brown dog.\n

WINDOWS

See the quick\r\n red fox jump\r\n over the lazy\r\n brown dog.\r\n

TEXT VS. BINARY FILES

• if (c == '\n')

Ē

- Windows (text files)
 - Input: $\r \wedge n \rightarrow \n$
 - Output: $n \rightarrow r n$
 - What about binary data (e.g., images, audio, video)?
 - Opens files in text mode by default

• ifstream in("filename", ios::binary);

ifstream in;
in.open("filename", ios::binary);