CSc 352
Binary File IO

Benjamin Dicken

Announcements

- Student Course Survey
  - 1 PA grade dropped if response percentage gets to 80% or more
  - 50.53% (as of before class)
- PA 10
  - Rectangles!
  - Need to fix **Object3D_append_quadrilateral**
- For PA 9, will try to give points for test cases that failed only due to rectangle issues

Fix the program

- COPY the files in **`/tmp/352cptest`** to your home directory

```
$ cp -r /tmp/352cptest ~/
```

- Compile the code with **make**
- What do you see?
- How can you fix it using only the preprocessor?

File Content

- Recall that files on a UNIX system are iNodes, that have pointers to data blocks, where the actual data of a file is stored
- Those blocks are just a bunch of 1’s and 0’s

- We can choose how to interpret when we read
- We can choose the format when we write

File Content

- Many of the files we have dealt with on UNIX in this course have been “text” files
    - *.c *.py *.txt *.stl makefile
    - This is just because we wrote text to those, and used programs that interpret files as text (vim)
- What have we used that are *NOT* text files?
- A “binary file” is just a file that we treat as information represented in RAW binary, rather than ASCII characters

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

int main() {
  uint32_t number = 10000000;

  FILE* text = fopen("text", "w");
  fprintf(text, "%", number);
  fclose(text);

  FILE* binary = fopen("binary", "wb");
  fwrite(&number, 1, sizeof(number), binary);
  fclose(binary);

  return 0;
}
```

What is this?

What is this?

Tools for viewing file contents

```
$ hexdump file_name

$ xxd -b file_name
```

Activity

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

int main() {
  uint64_t number = 20;

  FILE* text = fopen("text", "w");
  fprintf(text, "%lu", number);
  fclose(text);

  FILE* binary = fopen("binary", "wb");
  fwrite(&number, 1, sizeof(number), binary);
  fclose(binary);

  return 0;
}
```

Which file represents the number more efficiently?

Activity

Data Representation

Each row represents:
  studentID, exam 1, exam 2, final exam

How many bytes would it take to represent this with a CSV ASCII file?

How many bytes would it take to represent this in binary? How compact could we get it?

`grade_info.csv`

```
19311233,80,90,100
91246834,75,85,82
21245122,43,76,87
18673124,90,75,90
```

Activity

Implement Conversion

Write the code to:

- Open this text file
- Re-write the same data to **`binary_grade_info.bin`**
- Close the file

`grade_info.csv`

```
19311233,80,90,100
91246834,75,85,82
21245122,43,76,87
18673124,90,75,90
```

ELF

Executable and Linkable Format

The standard binary executable format for UNIX systems on x86 processors

A type of binary file!

Investigate ELF Files

```
$ file a.out

$ ldd a.out

$ readelf -h a.out

$ objdump -d a.out
```