Grading Guidelines

- 10% major bug
  - segfault
  - memory leak
  - major feature not working
- 5% minor bug
- 2-3% cosmetic bug
- Less deduction if you know about a bug, even if you couldn’t fix it

• 4.0 >= 95%
• 3.0 >= 75%
• 2.0 >= 50%

Printing

- Please use formatted printing to print code
  - Emacs
    - "Print Buffer"/"Postscript Print Buffer"
    - "M-x print-buffer"/"M-x ps-print-buffer"
  - Xcode (print from menu)
- Don’t create PDFs by cut-and-paste
  - It takes time and never looks as good
- If you ssh
  - "C-u M-x ps-print-buffer"
  - ps2pdf

Problems with lab machines

- Generate tickets at: systems.cs.brynmawr.edu
- If urgent, send email to ianj@cs.brynmawr.edu
- Or run to his office if you fork-bombed
- Telling Ian is more effective than telling me (I have to go to him anyway)

Source Code Organization

- Split into multiple source files
  - ll.c and hw1.c
- Every .c should have its own .h
- Might be useful to have an additional common.h
- Any operation on your data structure (i.e. linked list) outside of ll.c should be done via a function call provided through ll.h

Avoid Circular Includes

- Always enclose your .h with these directives:
  #ifndef NAME_H
  #define NAME_H
  /* contents */
  #endif
Avoid Unnecessary `printf`

- Sometimes it’s necessary to inform the user
- Otherwise, have a way to disable all debugging output in the finished product

```c
#if (DEBUG)
  //printfs
#endif
$(CC) $(CFLAGS) -DDEBUG -c hw1.c
```

Arrays/Lists Heap/Stack

- Use arrays unless you don’t know the size ahead of time and it changes constantly
- malloc an array if size is only known at run time
- Keep to automatic storage if you can — structs, arrays

```c
int arr[n];
int *arr = malloc(sizeof(int)*n);
```

- Automatic storage lives on the stack
- Dynamic storage lives on the heap
- Dynamic storage must be freed

No Integer Constants

- `#define` them
  - `#define FALSE 0`
  - `#define TRUE 1`
- Use `enum`
  - `enum{FALSE,TRUE};`
  - `enum{BLACK,RED,GREEN,BLUE};`
  - `enum{LINE,TRI,RECT,CIR};`
- Use `#defines` in system header files
  - `_NSIG //signal.h`

Things to Note

- Compile with `--Wall`
- Fix the warnings, do not ignore them
- Your Makefile should have the following targets:
  - `make` (compile to default target)
  - `make clean` (removes all .o and executable)
- Comment your globals
- Comment your function parameters
- Do not submit deep directory structures — call the "submit" script one single level up from where your code is

Memory and Pointers

- `char *str;`
- `char str[6];`
- `char *str = "CS355";`
- `char str[] = "CS355";`
- `char *str = malloc(sizeof(char)*6);`

hw1 specifics

- `char buff[SIZE];`
- `fgets/ read - use SIZE-1`
- `scanf("intr %d", &interrupts);`
- `open(outfd, O_WRONLY|O_CREAT|O_TRUNC, 0666);`
- `creat`
- `umask(S_IWGRP|S_IWOTH); // 022`