CS 354 - Machine Organization & Programming
Tuesday, October 1, 2019

Midterm Exam - Thursday, October 3rd, 7:15 - 9:15 pm

- **Lec 1 (2:30 pm):** room 3650 of Humanities
- **Lec 2 (4:00 pm):** room B10 of Ingraham Hall
- UW ID required
- #2 pencils required
- closed book, no notes, no electronic devices (e.g., calculators, phones, watches)
- see "Midterm Exam 1" on course site Assignments for topics

Project p2B (3%) DUE: 10 pm, Monday, October 7th

Homework hw2 (1.5%) DUE TOMORROW: 10 pm, Wednesday, October 2nd

Last Time

Pointers to Structures
Standard & String I/O and `stdio.h`
File I/O and `stdio.h`
Copying Text Files

Three Faces of Memory
Virtual Address Space
C's Abstract Memory Model

Today

C's Abstract Memory Model (from last time)
Meet Globals and Static Locals
Where Do I Live? (from last time)
Linux: Processes and Address Spaces
---------- END of Exam 1 Material ----------
Meet the Heap

Next Time

Exam Mechanics
Heap Allocator Design

Meet Globals and Static Locals

What?

A *global variable* is

-
-
-

A *static local variable* is

-
-
-

Why?

❋ *In general, global variables*

How?

```
#include <stdio.h>
int g = 11;

void f1(int p) {
   static int x = 22;
   x = x + p * g;
   printf("%d\n", x);
}

int main(void) {
   int g = 1;
   f1(g);
   g = 2;
   f1(g);
   return 0;
}
```

shadowing:

❋ *Don't use the same identifier for local variables*

Linux: Processes and Address Spaces

Process and Job Control

-

```
ps

jobs

&
```
ctrl+z

```
bg
fg
```

ctrl+c
```
kill

top
```

Program Size

size <executable or object_file>

```
$gcc -m32 myProg.c
$size a.out
   text    data     bss     dec     hex filename
   1029     276       4    1309     51d a.out
```

Virtual Address Space Maps

-

```
$pmap <pid_of_process>

$cat /proc/<pid_of_process>/maps

$cat /proc/self/maps
```

/proc:

```
$cat /proc/loadavg
```

Meet the Heap

What? The heap is

- *dynamically allocated memory*:

- *block*:

  payload:

  overhead:

  allocator:

Allocator Approaches

Implicit:

-

-

Explicit:

-

-