malloc, free, Memory, string.h

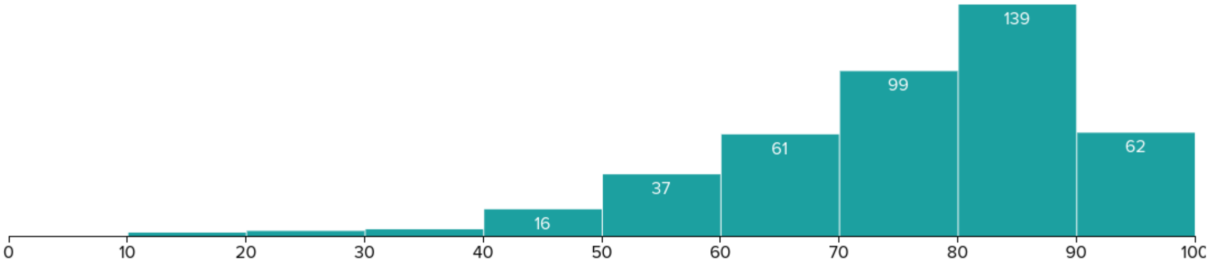
CS 2130: Computer Systems and Organization 1

November 16, 2022

Announcements

- Homework 8-10 posted, due last day of class at 11pm
- Lab 10 please try on your own (with classmates), check off by last day of class
- No more quizzes this semester

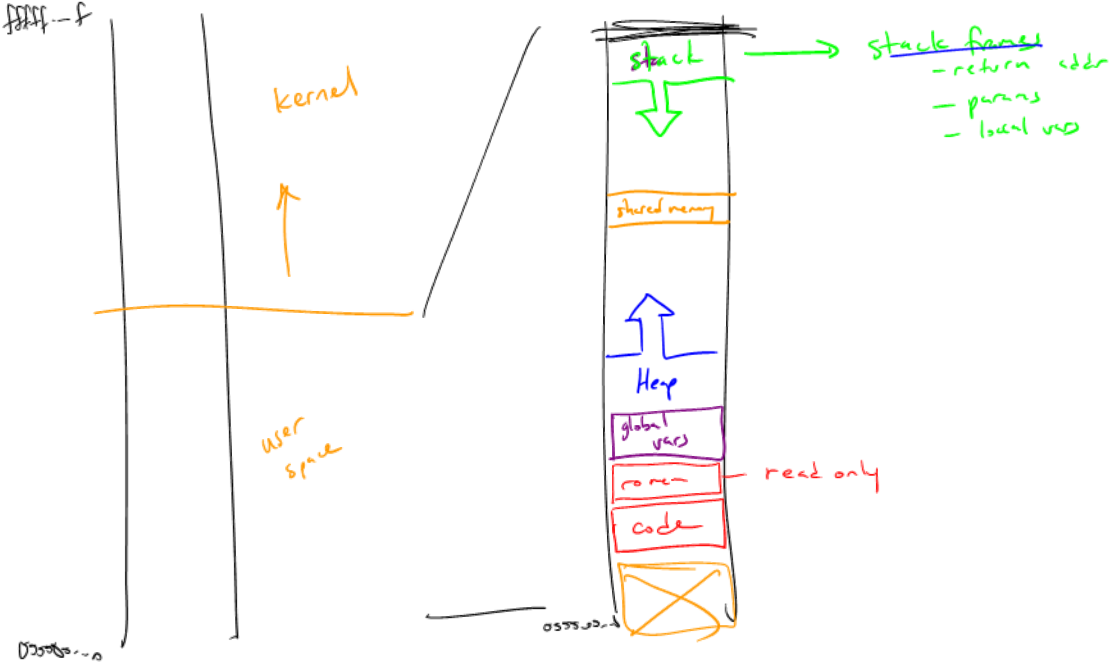
Exam 2



Statistics

	Exam 1	Exam 2
Mean	75.4	75.5
Median	77.0	79.0
Std. Dev.	15.4	14.6

Memory


ffff...f
kernel
user space
0000...0
stack
stack frames
- return addr
- params
- local vars
shared memory
Heap
global vars
ro mem
read only
code
0555...0

Memory

An Interesting Example

```
int *makeArray() {
    int answer[5];
    return answer;
}

void setTo(int *array, int length, int value) {
    for(int i=0; i<length; i+=1)
        array[i] = value;
}

int main(int argc, const char *argv[]) {
    int *a1 = makeArray();
    setTo(a1, 5, -2);
    return 0;
}
```

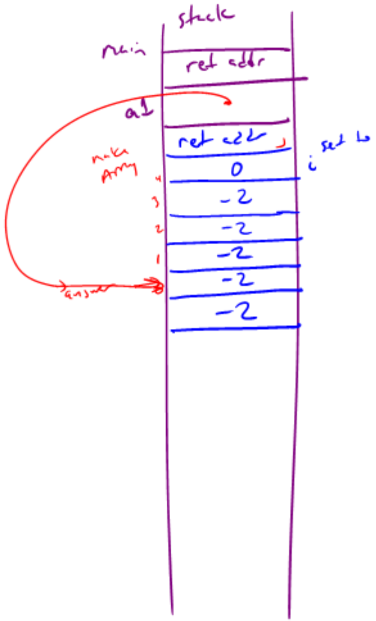
— escaping pointer

5

-2

rsi

rdx



The Heap: Requesting Memory

```
void *malloc(size_t size);
```

- Ask for `size` bytes of memory
- Returns a `(void *)` pointer to the first byte
- It does not know what we will use the space for!
- Does not erase (or zero) the memory it returns

malloc Example

```c
typedef struct student_s {
    const char *name;
    int credits;
} student;

student *enroll(const char *name, int transfer_credits) {
    student *ans = (student *)malloc(sizeof(student));
    ans->name = name;
    ans->credits = transfer_credits;
    return ans;
}
```

The Heap: Freeing Memory

Freeing memory: `free`

```
void free(void *ptr);
```

- Accepts a pointer returned by `malloc`
- Marks that memory as no longer in use, available to use later
- You should `free()` memory to avoid *memory leaks*

Common Memory Bugs (reading)

List example