C, Memory

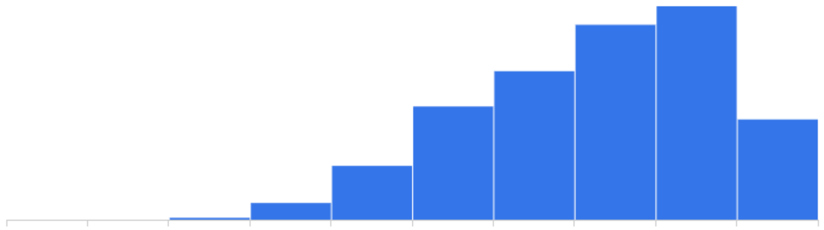
CS 2130: Computer Systems and Organization 1

April 14, 2023

Announcements

- Homework 8 due Monday at 11pm
    - Limited number of submissions, test your code before submitting
- Quiz 8 opens today, please submit before 11:59pm Sunday
- Exam scores out later today, regrade requests by next Friday

Exam 2



Mean	72.1
Median	74.0
Std Dev	14.75

header example

`string.h`

variadic functions

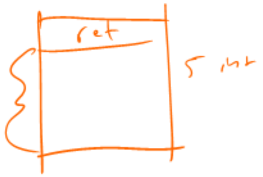
Memory

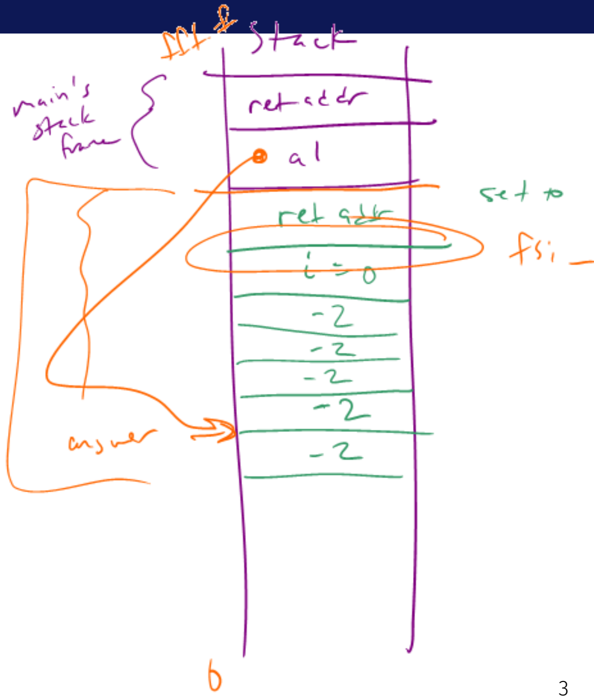
An Interesting Stack Example

```
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;
}
```





The Heap

The heap: unorganized memory for our data

- Most code we write will use the heap
- *Not a heap data structure…*

The Heap: Requesting Memory

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

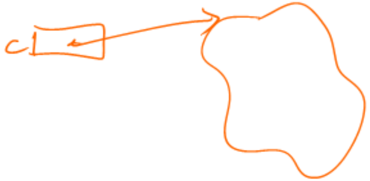
- Ask for `size` bytes of memory
- Returns a `(void *)` pointer to the first byte
- It does not know what we will use the space for!
- Does not erase (or zero) the memory it returns

Java

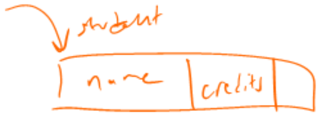
What is the closest thing to **`malloc`** in Java?

MyClass c = new MyClass();

new int[100]



`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;
}
```

(*ans).name

(*ans).credits

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*

Garbage

Garbage - memory on the heap our code will never use again

- Weird: defined in terms of the future!
- Compiler can't figure out when to free for you

Garbage

Garbage - memory on the heap our code will never use again

- Weird: defined in terms of the future!
- Compiler can't figure out when to free for you

What about Java?

Garbage Collector

Garbage Collector - frees garbage “automatically”

- **Unreachable memory** - memory on heap that is unreachable through pointers on the stack (or reachable by them)
  - Subset of all the garbage
  - Identifiable!
- Takes resources to work
- *Very* popular - most languages have garbage collectors
  - Java, Python, C#, …

`malloc` man page

List example

Common Memory Bugs (reading)