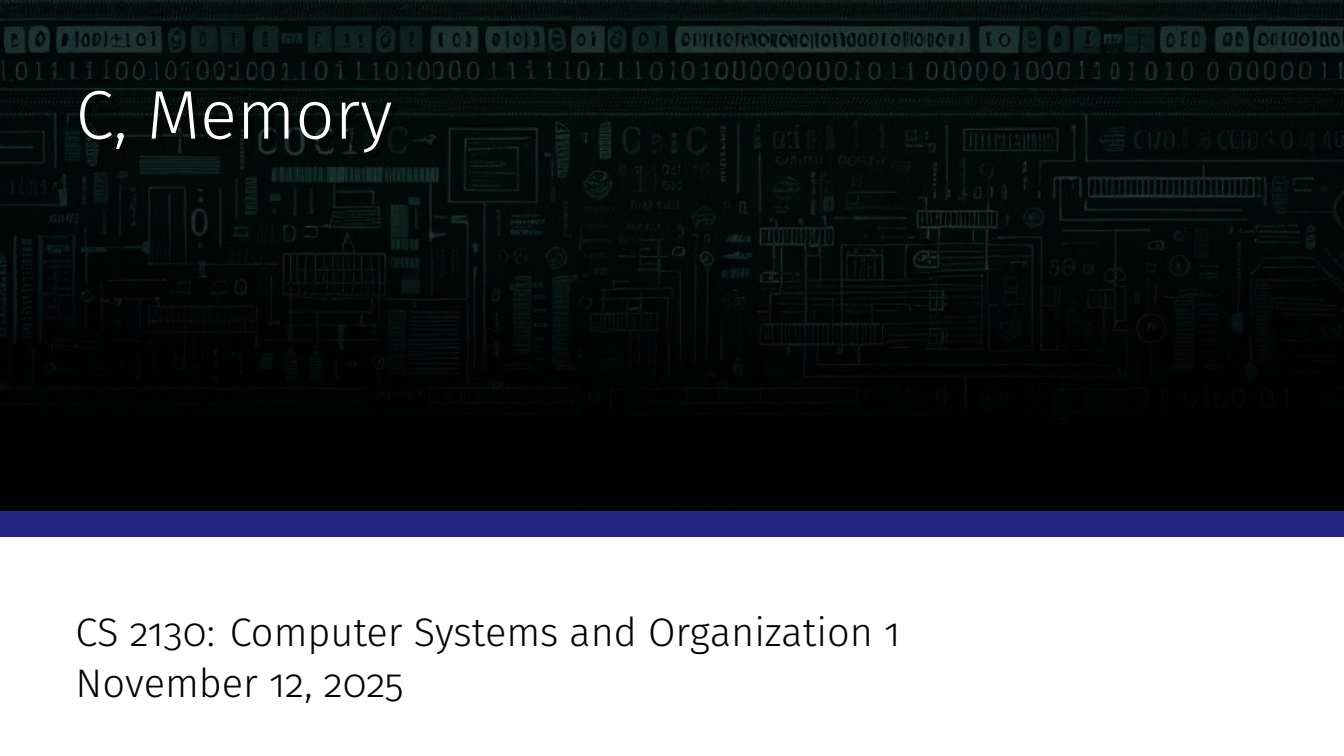




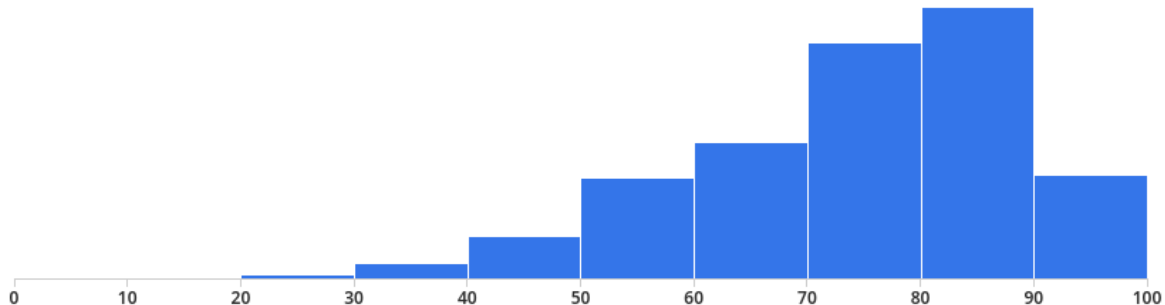
C, Memory

CS 2130: Computer Systems and Organization 1
November 12, 2025

Announcements

- Homework 8 due Monday on Gradescope
- Quiz 8 out today, due Sunday night

Exam 2



Median

77.0

Maximum

100.0

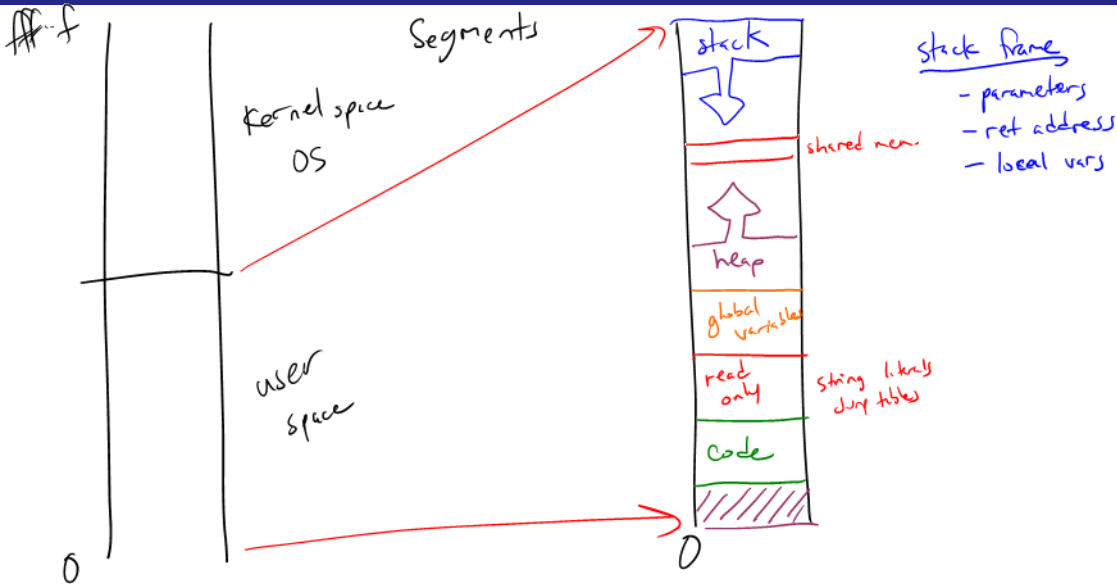
Mean

73.88

Std Dev [?](#)

14.51

Memory



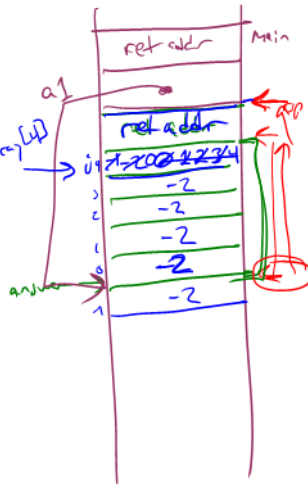
An Interesting Stack Example

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

escaping pointer

```
void setTo(int *array, int5 length, int-2 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

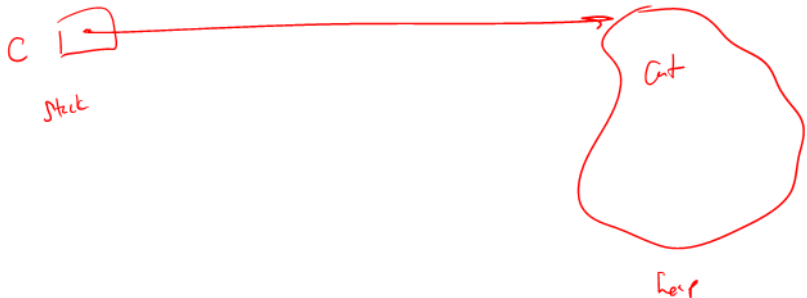
```
void *malloc(size_t size);
```

- 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?

`new int[100]`
`Cat c = new Cat("Aido");`

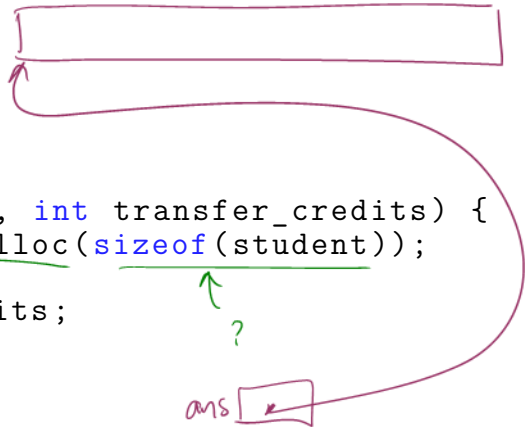


malloc Example

```
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 -> name



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

`calloc`, `realloc`

Common Memory Bugs (reading)

List example