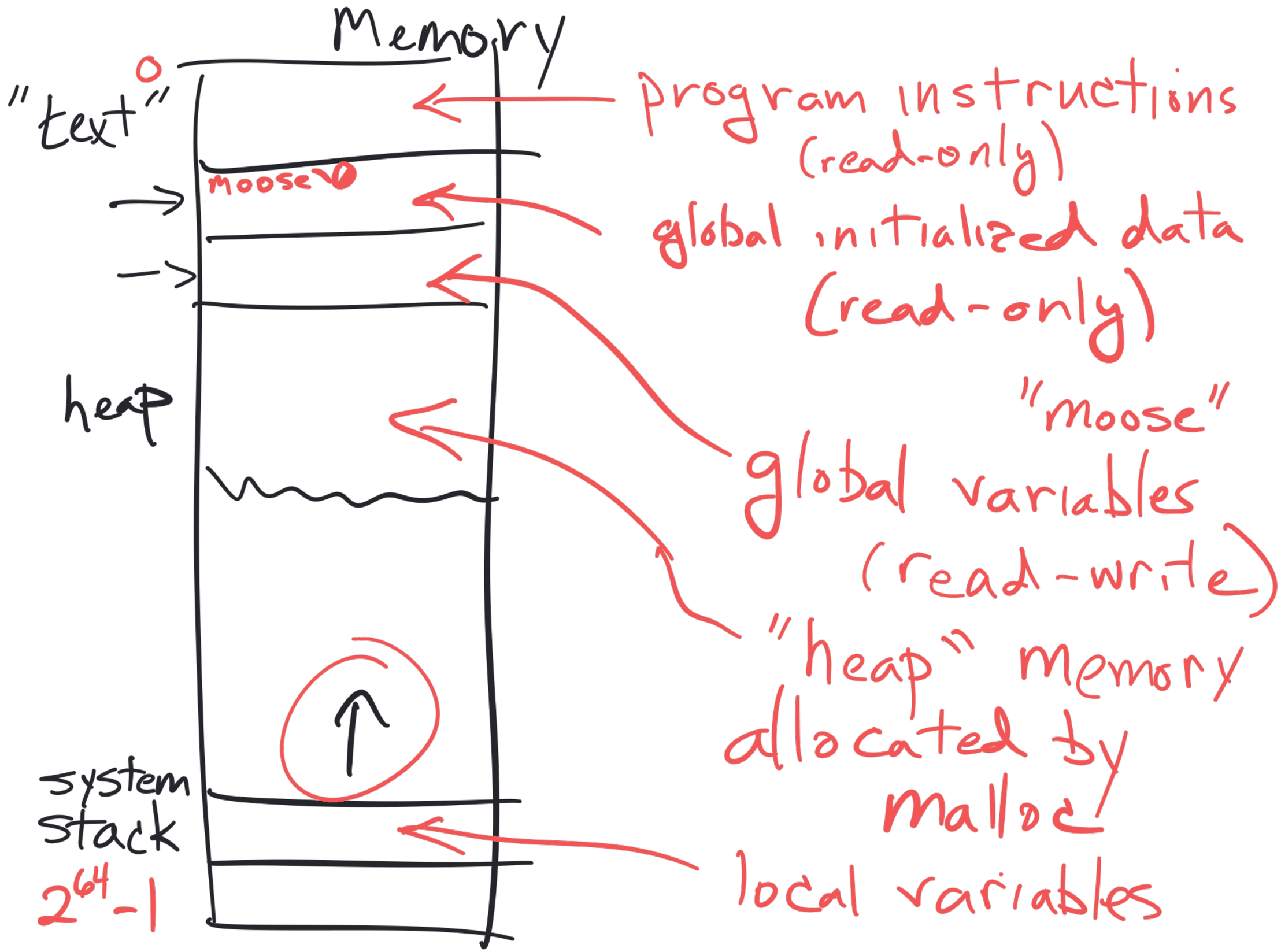
CS 208

Wed, 11 Jan 2023


Memory
0
"text"
moose
heap
system
stack
$2^{64}-1$
program instructions
(read-only)
global initialized data
(read-only)
"moose"
global variables
(read-write)
"heap" memory
allocated by
malloc
local variables

Variables

```
int j;
```

① "Declaration" there is a variable named j of type int

② "Definition" — hey run-time system, set aside 4 bytes to hold j (sizeof(int))

int j = 5;

Declare (int, j)

Definition (4 bytes allocated)

Initialization (put 5 into those 4 bytes)

& operator (unary operator)

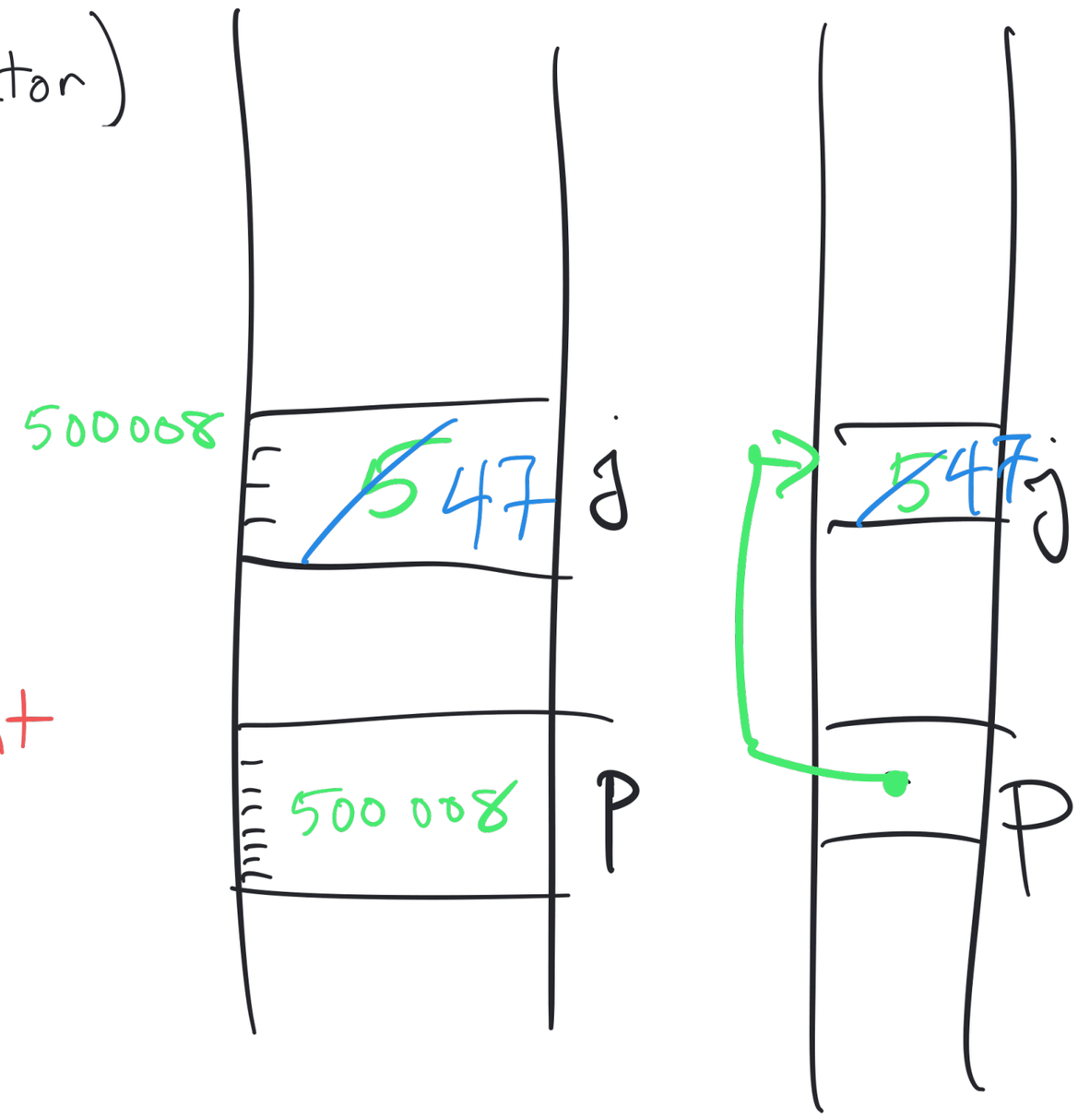
"the address of"

```
int j = 5;
int *p = &j;
```

↑ "pointer to" an int
"address of" an int

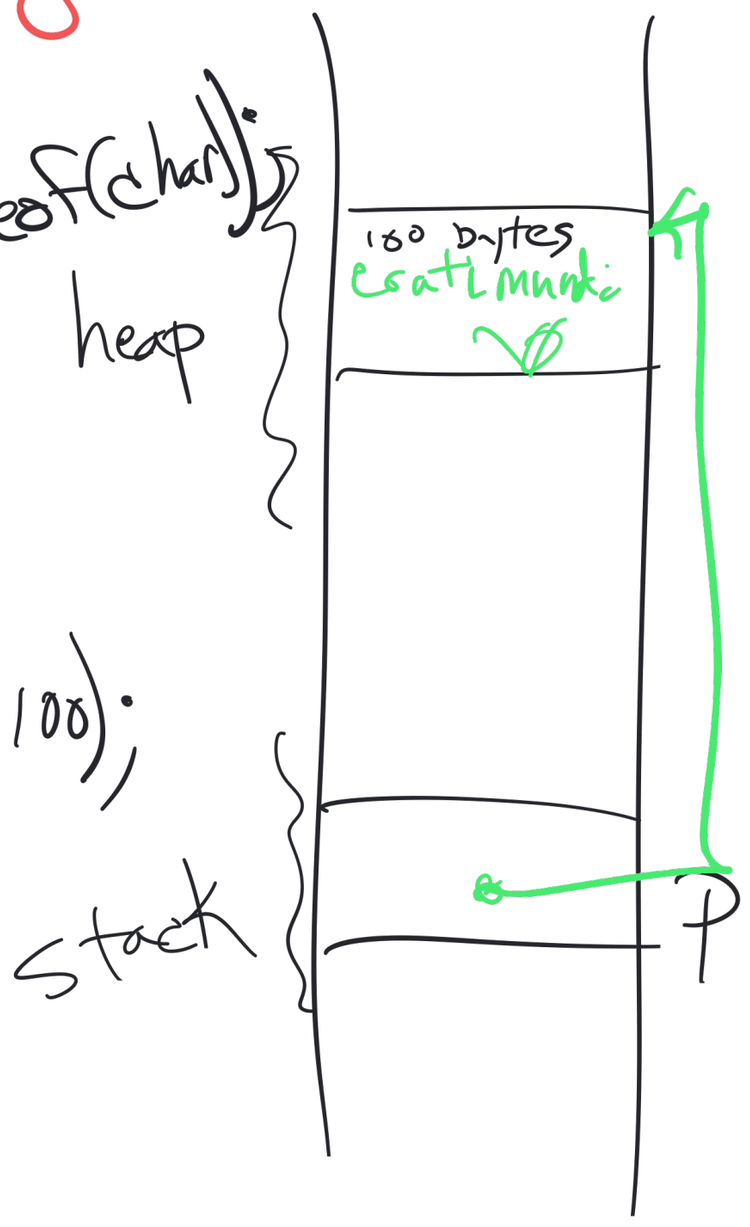
```
*p = 47;
```

↑ dereferencing p



typecasting

```
char *p;
p = (char *)malloc(100 * sizeof(char));
if (p == NULL) {
        uh-oh
        exit(1);
}
strncpy(p, "coatimundi", 100);
   .
   .
   .
// Done w/ p?
free(p);
```



typedef unsigned int size;

old type

new type

size j;

unsigned int j;

} identical

```
struct circle {
        int center_x;
        int center_y;
        int radius;
};
```

```
struct circle c;
c.center_x = 5;
c.radius = 23;
```

```
typedef struct circle {
        int center_x;
        int center_y;
        int radius;
} circle;
```

old type

new type

```
circle c;
c.center_x = 5;
```

```
circle  c;
circle *p = &c;
(*p).center_x = 5;
 p->center_y = 19;
```