Lecture 17

Thursday, October 20, 2022 11:15 AM

ANSI C99

static allocation → global space — size known @ compile time — @ compile time

automatic allocation → stack — live until function returns — @ runtime (VLAs)

Function returns something of size known only @ runtime?

dynamic memory allocation → heap

size: chosen at runtime (sometimes could fail)

lifetime: when allocated → when freed

```
void * malloc (size_t size);   - allocates size bytes
sizeof                         - operator - type - sizeof(struct)
                                           - expression sizeof(me)
void free (void *);            - pass a pointer from   sizeof(me.exam_scores)
```

malloc

```
void * realloc (void *ptr, size_t size);
```

- no longer have to free ptr, but you do have to free the return value

clear

```
void * calloc (size_t count, size_t size);
```

all allocators return NULL if there is not enough space

size_t — some unsigned int type.

```
int x[40];
int *p;
:
for (i=0; i < sizeof(x)/sizeof(x[0]); i++)
```

p p[0]

```
int x[4][5];
int *p = &x[0][0];
```

[i*5+j]

invalid for i ≠ 0

x[0][i*5+j];

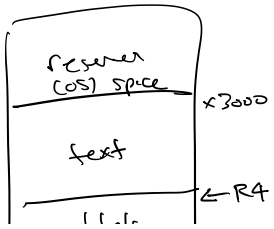
p[i*5+j]

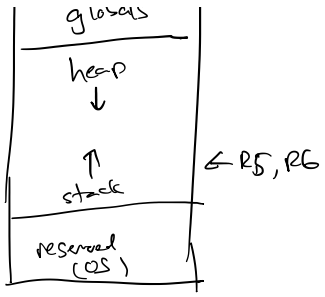
Valid (only way)

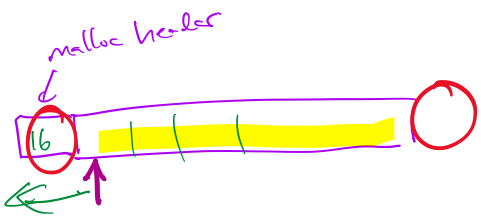
dangling pointers — memory that has been freed (possibly reallocated)

garbage — space that is no longer used but not freed

across control flow, know who is responsible for freeing pointer







ElectricFence

Valgrind