CS 208

F, 26 Jan 2024

```
char p[10] = "goat";
```

① declaration: name + type of a variable

② memory allocation: (here) set aside ("definition") aside 10 bytes for 10 chars.

③ Initialization: put some data into those bytes.

char p[10];

just declaration + mem alloc

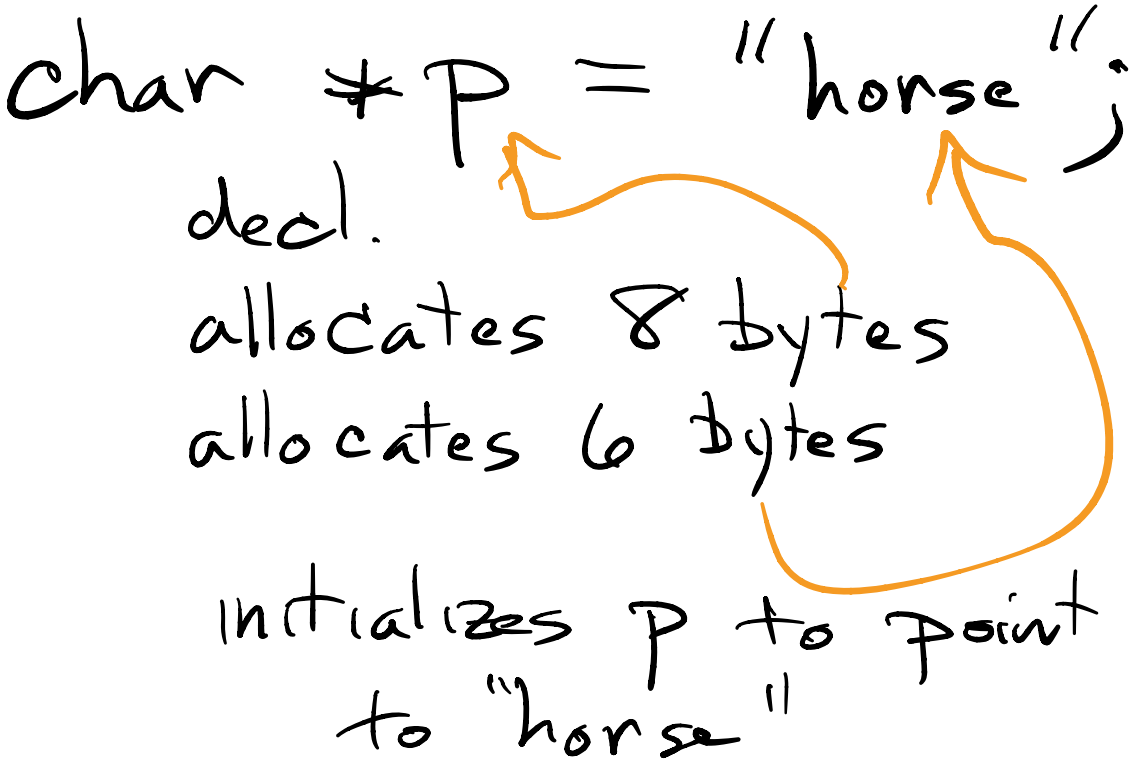
---

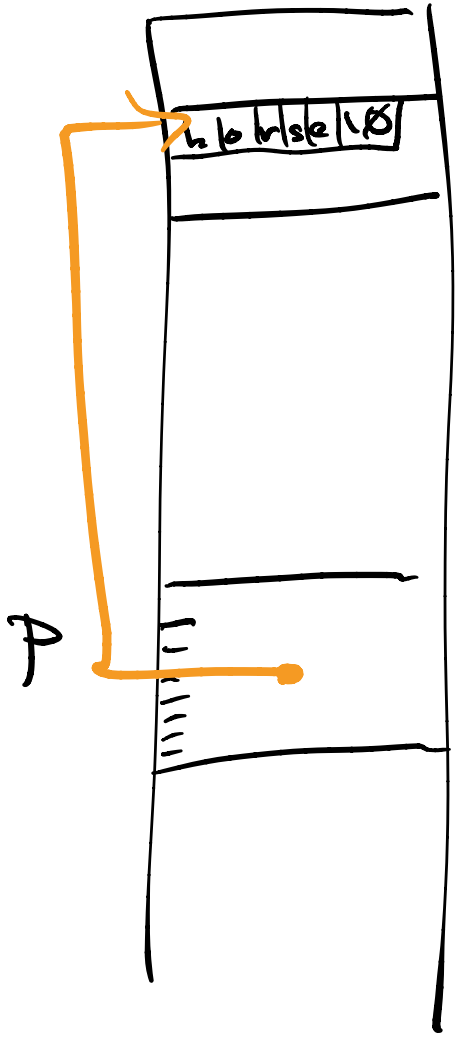
char *p;

declaration + mem alloc

8 bytes to hold the pointer

no initialization yet.


char *p = "horse";
decl.
allocates 8 bytes
allocates 6 bytes
initializes p to point to "horse"



literal strings

```
char *p = "horse";
```

read-only memory

```
strcpy(p, "goat");
```

crashes

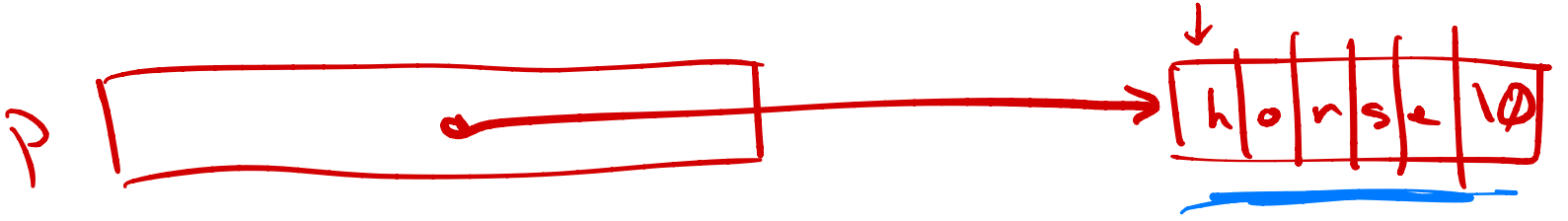
you're trying to write g o a t \0 to read-only memory

```
char *p = "horse";
printf("%c", p[2]);
```

→ r on screen

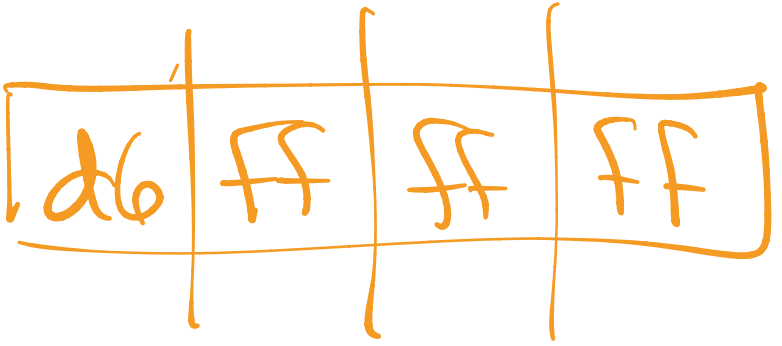
compiler arranges:

```
*(p + 2*sizeof(char))
```



Quiz #4

k



little-endian

k == 0xffffffd6

What do I add to this to get zero

0xffffffd6 - 32-bit 2's comp

-42

1111 1111 1111 1111 1111 1111 1101 0110

negative

0000 0000

\+

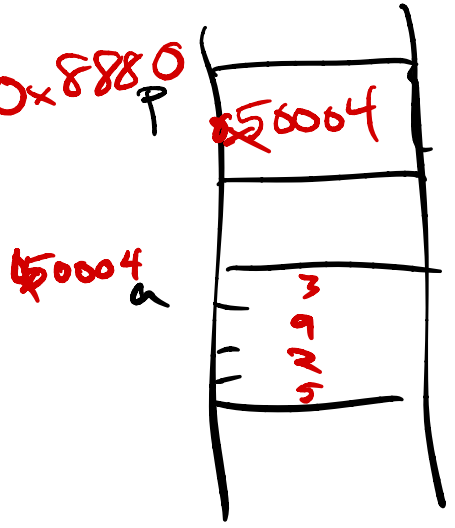
10 1010

0 0

32 + 8 + 2 = 42

More pointer stuff

```
int a[4] = {3, 9, 2, 5};
int *p = a;
```




```
printf("%ox", a) → 50004
printf("%ox", p) → 50004
printf("%ox", a[0]) → 3
printf("%ox", &p) → 8880
printf("%ox", *a) → 3
```

int a[4];

sizeof(a) — ~~4~~ 16

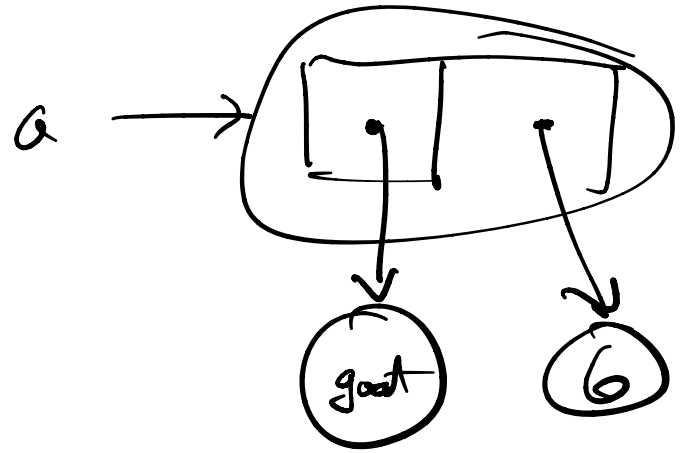
4 * sizeof(int)

int *p = a;

sizeof(p) — 8?

Mixed-type arrays?

Python `a = ['goat', 6]`



Not in C

`void *a[100];`

```
int b = 0x08005432;
char c = (char)b;
```

c ← 0x32