2510 : Lecture 18 || March 3, 2017

Assignment 2 comments include:
- email, date, name, student number
- brief description of program

Structures

- allows us to package items of different types together

eg.

```
struct score {
    char id[10];
    int score;
};
```

members of the structure

/* structure definition: defines a structure type */

semicolon needed

how we use Structures:

```
struct score s;
```

like int n;

type (like int)

```
s.score = 85;
strcpy(s.id, "a1111 1111");
```

eg.

```
struct score a[10], *p;
a[0].score = 78;
strcpy(a[0].id, "A22222222");
a[1] = a[0];
p = &a[1];
printf("%d\n", (*p).score);
```

like int a[10], *p;

assignment of structures

p points to structure

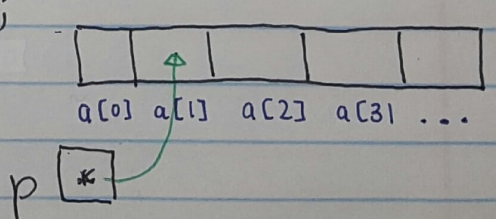
address of

necessary b/c precedence

↳ p->score

*p.score ≡ *(p.score)

doesn't make sense in this case

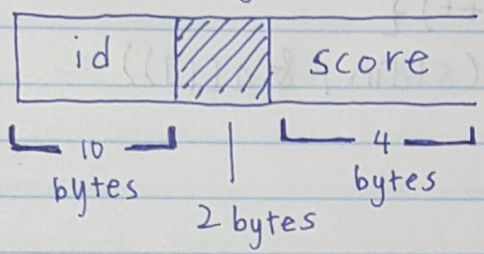


∴ a -> b ≡ (*a).b

→ In general, sizeof(a structure) >= sum of sizes of its members.

↳ the compiler is allowed to add paddings between members at the end of a structure due to alignment requirements.

↳ possible layout of struct score:



written again below (sry its messy)

▷ Nested Structures

```
struct Name {
    char first[20];
    char last[20];
};
struct Grade {
    char id[10];
    struct Name name;
    int score;
};
```

function to read grade

```
int read_grade(FILE*fp,
    struct Grade *g){
return fscanf(fp,"%s%s%s%.,
    g->id, g->name.first, g->name.last,
    &g->score) == 4;
```

note: no validation, assume input is valid except EOF.

```
struct Grade c2510[120];
strcpy(c2510[0].id, "A66666666");
strcpy(c2510[0].name.first, "monty");
strcpy(c2510[0].name.last, "burns");
c2510[0].score = 15;
```

cont...
```
int read_grade(FILE *fp, struct Grade *g){
  return fscanf(fp, "%s%s%s%s", g->id, g->name.first,
    g->name.last, &g->score) == 4);
}
```

```
struct Grade c2510[120];
size_t i;
for(i=0; i<120; i++){
   if (!read_grade (stdin, &a[i]))
         break;
}
```

▷ type def : used to give a type another name.

eg. `unsigned long long x;`

`type def unsigned long long (x);` — x: variable name of ull; type unsigned long long

eg.
```
Struct Grade grade;
type def struct Grade grade;
grade g, a[10], *p;
```
grade ≡ Struct Grade

Create a type set that stands for 25 ints

eg. `set a, b, c, d, e;`

`type def int set[25];`