

数据 (Data): 结构体, 数组.

机器指令没有内建的聚合数据类型.

数组: $T \ A[i]$: 收集 $L \times \text{sizeof}(T)$ 字节的数组

A 可以用作指向下标为 0 内元素的指针.

附: 在 C 语言中, 对于 $T * p$, 进行自增

$p++$ 等效于 $p += \text{sizeof}(T)$, 而 $p+i$

等效于 $p + \text{sizeof}(T) \times i$, 同样适用于 $i \leq 0$

的情况, 注: 不能写作 $i+p$

| Decl | A_n | | | $*A_n$ | | |
|-----------|-------|-----|------|--------|-----|------|
| | Cmp | Bad | Size | Cmp | Bad | Size |
| int A1[3] | Y | N | 12 | Y | N | 4 |
| int *A2 | Y | N | 8 | Y | Y | 4 |

区别: 数组与指针.

例 2.

| Decl | A_n | | | $*A_n$ | | | $**A_n$ | | |
|--------------|-------|-----|------|--------|-----|------|---------|-----|------|
| | Cmp | Bad | Size | Cmp | Bad | Size | Cmp | Bad | Size |
| int A1[3] | ✓ | ✗ | 12 | ✓ | ✗ | 4 | ✗ | ✓ | ✗ |
| int *A2[3] | ✓ | ✗ | 24 | ✓ | ✗ | 8 | ✓ | ✓ | 4 |
| int (*A3)[3] | ✓ | ✗ | 8 | ✓ | ✓ | 12 | ✓ | ✓ | 4 |
| int (*A4[3]) | ✓ | ✗ | 24 | ✓ | ✗ | 8 | ✓ | ✓ | 4 |

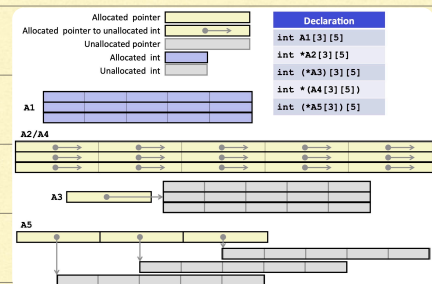
均为指针
数组
指向指针数组的
指针

例 3:

| Decl | A_n | | | $*A_n$ | | | $**A_n$ | | |
|-----------------|-------|-----|------|--------|-----|------|---------|-----|------|
| | Cmp | Bad | Size | Cmp | Bad | Size | Cmp | Bad | Size |
| int A1[3][5] | | | | | | | | | |
| int *A2[3][5] | | | | | | | | | |
| int (*A3)[3][5] | | | | | | | | | |
| int *(A4[3][5]) | | | | | | | | | |
| int (*A5[3])[5] | | | | | | | | | |

- Cmp: Compiles (Y/N)
- Bad: Possible bad pointer reference (Y/N)
- Size: Value returned by sizeof

| Decl | $***A_n$ | | |
|-----------------|----------|-----|------|
| | Cmp | Bad | Size |
| int A1[3][5] | | | |
| int *A2[3][5] | | | |
| int (*A3)[3][5] | | | |
| int *(A4[3][5]) | | | |
| int (*A5[3])[5] | | | |



高程数组:

引用 $A[i \times j]$ ($A[R][C]$), 会计算 $A + (i \times c + j) \times 4$

结构体:

- ① 结构体呈现为内存的一块区域
- ② 根据声明顺序分配内存.
- ③ 编译器决定使用的内存区域.

对齐: 保证每个元素的地址为最大数据类型时整数倍, 保证效率以及防止出现错误.

- 1 byte: char, ...
 - no restrictions on address
- 2 bytes: short, ...
 - lowest 1 bit of address must be 0₂
- 4 bytes: int, float, ...
 - lowest 2 bits of address must be 00₂
- 8 bytes: double, long, char *, ...
 - lowest 3 bits of address must be 000₂
- 16 bytes: long double (GCC on Linux)
 - lowest 4 bits of address must be 0000₂

以及向结构体的末尾添加字节来让结构体在外部对齐. (仅针对基本元素, 不针对集合元素).

fix: 将最大的元素放在开头, 再依次放较小.

浮点数: x87 专用, 极为丑陋

浮点寄存器: xmm 寄存器.

↓
SSE 使用向量指令
↓
AVX.