Data Structures

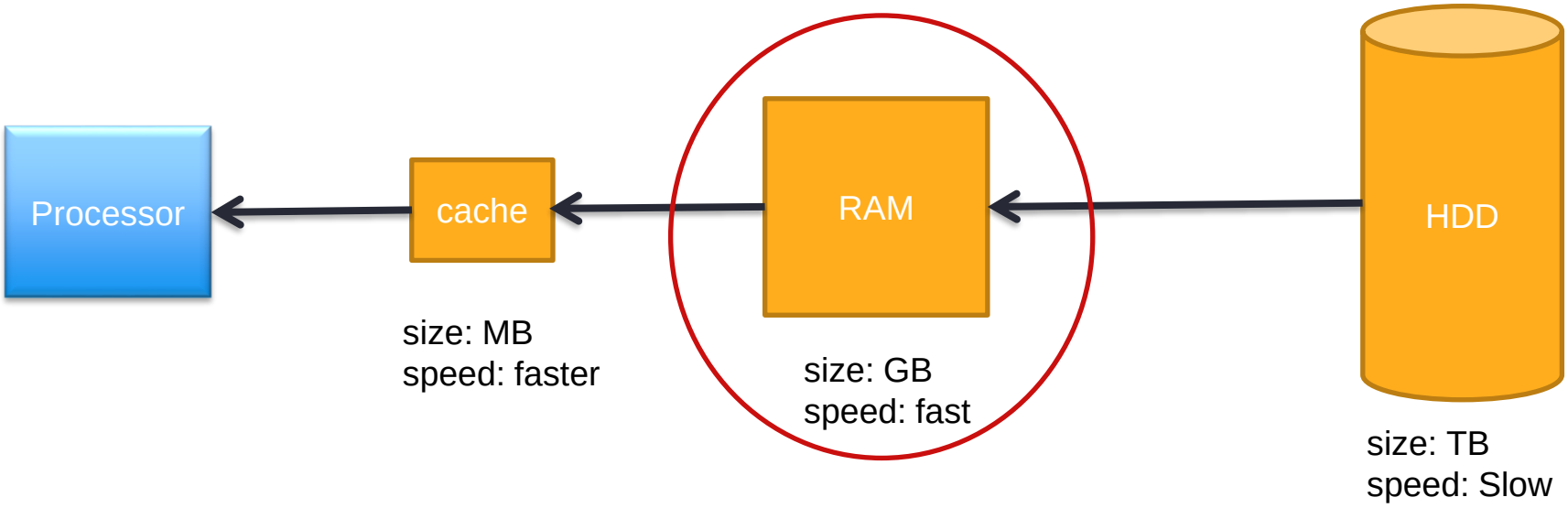
lecture 7

1-10-2022

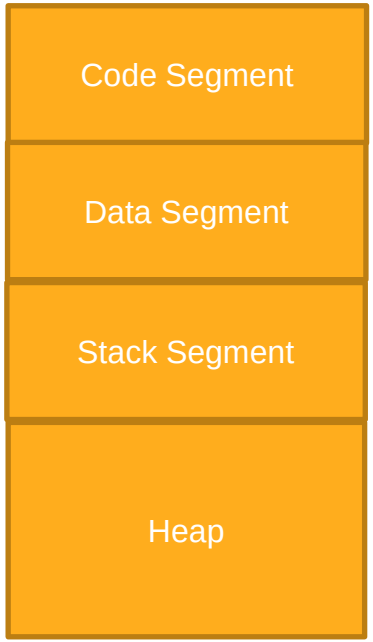
Last Session Quick Revision

Basics of Memory Management

Memory Types



Closer look to RAM



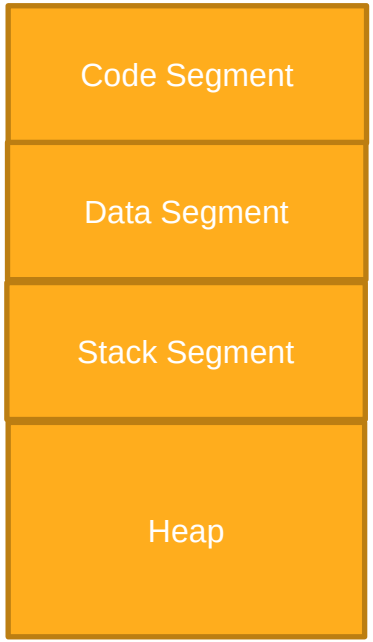
Code Segment:

- Stores plain statements
- Not useful for programmer from storage manipulation perspective

Data Segment

- Stores global and static variables
- Comparatively smaller

Closer look to RAM



- Stack Segment (SS)
  - Stores local variables (variables in function)
  - As function called local variables are inserted on ss
  - As function returns (last line of function definition executes) variables are removed from stack.
  - **Good Thing:**
    - Memory management automatic
  - **Bad Thing:**
    - Limited in size

A closer look to pointers

- **Pointer is a special variable which stores address of other variable.**

Regular variable

•Stores value

•data-type var_name;

•int x

Pointer Variable

•Stores address

•data-type* var_name;

•int* p

Dereferencing a Pointer

- Finding out value at the address stored in the pointer
- int v = 10;
- int* ptr = &v;
- printf(“%d”,ptr);
- printf(“%d”,*ptr);

Pointer to Pointer

- `int v = 10;` `//variable`
- `int* p1 = &v;` `//pointer to int variable`
- `int** p2 = &p1;` `//pointer to int* variable`

Pointers and Arrays

```
void main() {
    int a[ ] = {1,2,3,4,5}, *p;
    p = a;
    ++*p;
    printf("%d ", *p);
   p += 2;
    printf("%d ", *p);
   }
```

Dynamic Memory Allocation

- Refers to allocating memory on heap
- C uses special functions for it
  - malloc()
  - Calloc()
- Heap Memory mgmt is not automatic:
  - if you allocate memory on heap manually (malloc/calloc)
  - You must de allocate it manually (free)

Pointers to Structure

```
struct complex
    {
        int real;
        int img
    };
    struct complex c1;    //variable of type complex
    struct complex* ptr = &c1; //pointer to structure
```