Data Structures and Algorithms

Mohammad GANJTABESH

Department of Computer Science,
School of Mathematics, Statistics and Computer Science,
University of Tehran,
Tehran, Iran.

mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

Variables, Pointers and Arrays

C++

Data Structures and Algorithms
Undergraduate course

Mohammad GANJTABESH
mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

Data types

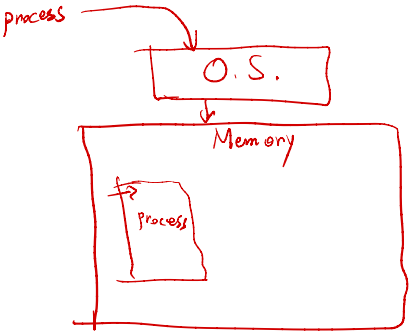
- Data type indicates the category of data and its properties (including the size, permitted operations, etc.).
- Different data types:
  - Built-in types : elementary data types in programming language (int, char, float ...)
  - Arrays : collection of data with identical type.
  - Struct and class : collection of data (methods) with possibly different types.
  - Data structures : User-defined collection of data and its operations.

Memory management

- How different types of data are stored and manipulated in memory?
- The role of operating system : managing all the resources (memory, etc.)
- Different parts of program/process: code, data, stack, heap

Memory management:

- Segmentation
- Paging



Memory management: segmentation

```
int main(){
    int x;
    char c;
    cin >> x >> c;
    cout << x << c;
    return(0);
}
```

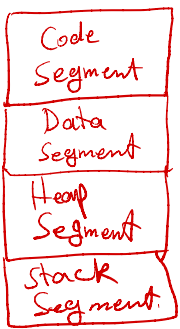
Compile

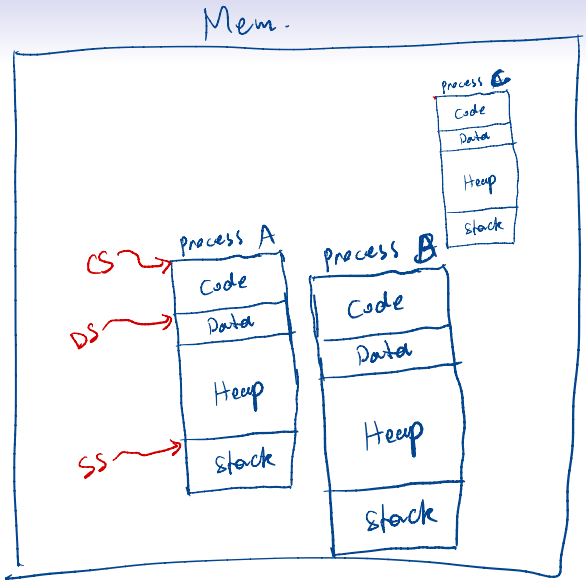
Machine code. (0/1)

File System

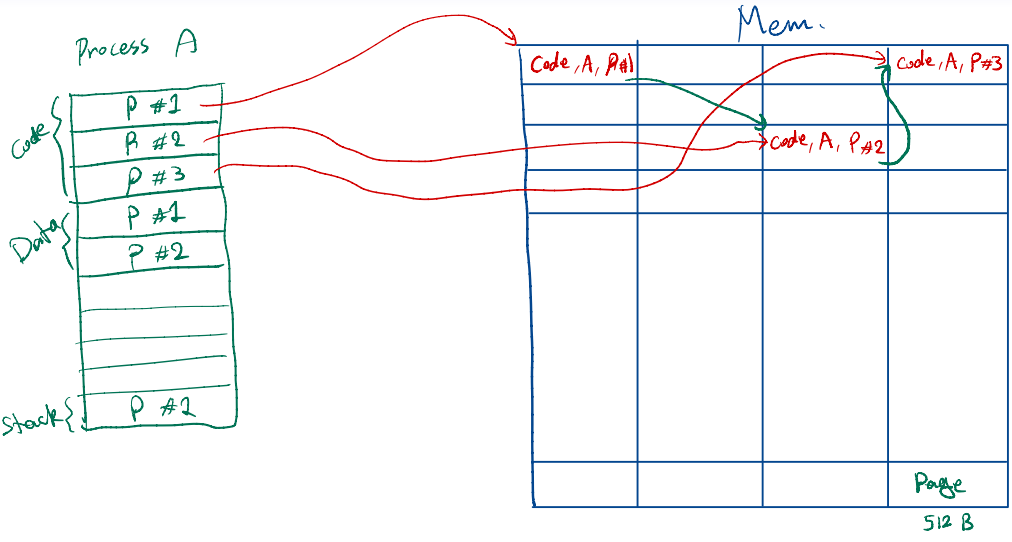
main.exe file

Process main



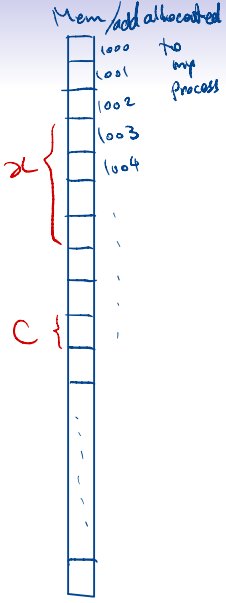

Mem.
process C
Code
Data
Heap
Stack
process A
CS
Code
DS
Data
Heap
SS
Stack
process B
Code
Data
Heap
Stack

Memory management: paging



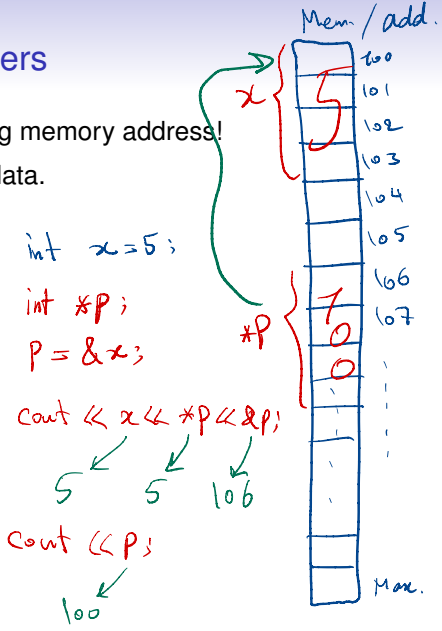
Variables

- A place-holder in memory for storing data.
- Variables' attributes:
  - The number of allocated memory bytes
  - The location of the allocated memory
  - The defined/permitted operations on it
  - Type casting: convert one type to another



Pointers

- A place-holder in memory for storing memory address!
- Can be used to access the stored data.

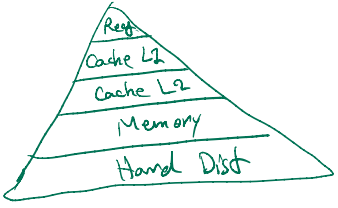


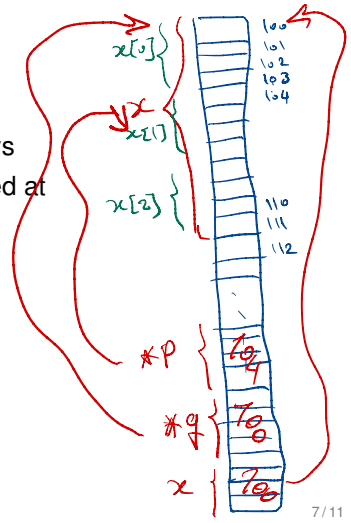
type *p;
p++; ≡ p = p + sizeof(type)

Arrays

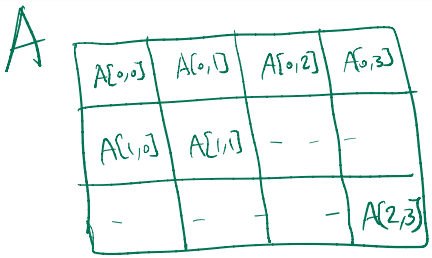
int x[3] = {1,5,7};
int *p = &x[1]; p++;
int *q = x;

- Contiguous area of memory consists of equal-size elements indexed by contiguous integers
- Random access (constant time)
- Fixed size (static memory allocation)
- Consecutive memory bytes are allocated to arrays
- Number of elements in an array could be specified at
  - compile time (static memory allocation)
  - run time (dynamic memory allocation)





Multi-dimensional arrays



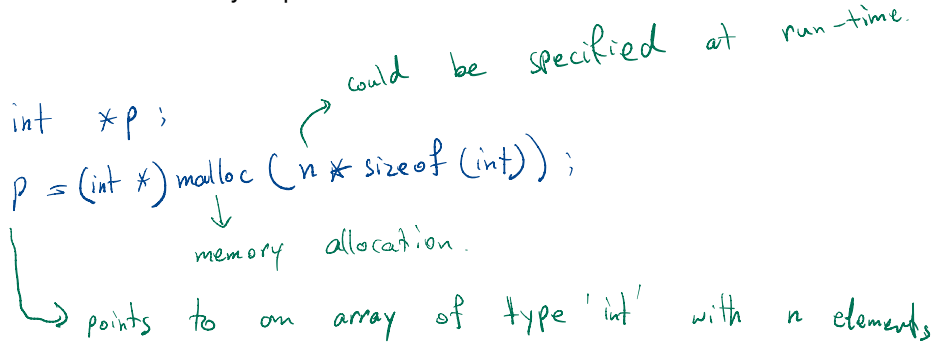
row-major

k = index of element $[i,j]$ in mem

$= A + (i * n) * \text{elementSize} + j * \text{element Size}.$

Dynamic memory allocation

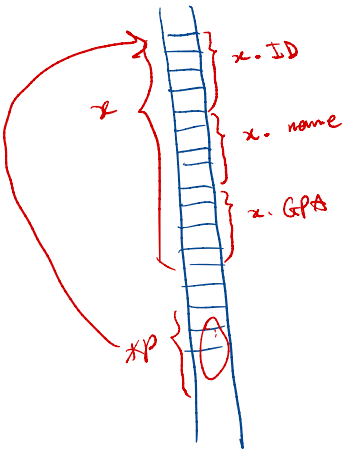
- Should be asked from the operating system to do the memory allocation at run-time.
- Pointers are usually required.



Struct and Class

- To define a new type containing different types of data (fields).
- User-defined type.

```
struct Student {
    int ID;          4 bytes
    char * name;     4 bytes
    float GPA;       4 bytes
    :
    :
};
Student x;
struct *p = &x;
```



Abstract Data Type (ADT)

Concerns what is getting done not how it is getting done.

Definition

A class of objects whose logical behavior is defined by a set of values and a set of operations.

- Constructors.
- Transformers.
  change/modify the state of ADT
- Observers:
  allow read-only access to the state of ADT
- Iterators:
  allow sequential access to the elements in ADT.

- Destructors.
  clear the state of ADT.