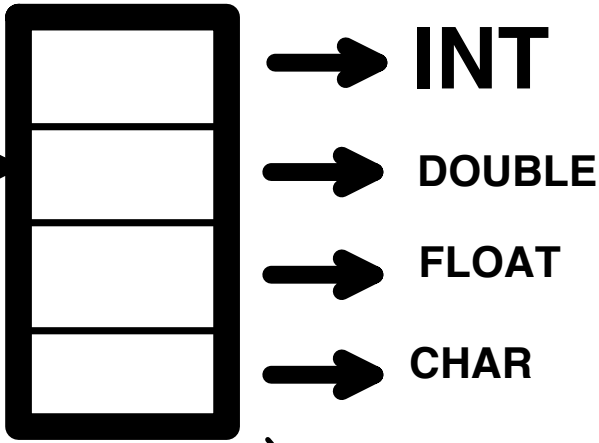


STRUCT

La struct permette di accorpare in un'unica area di memoria, informazioni di natura "eterogenea".

Graficamente all'interno del nostro address space abbiamo la possibilità di mantenere tipi di dati differenti.



un impiego classico è quello di creare una tabella di informazioni eterogenee in termini di tipologia e quantità di memoria richiesta per la loro rappresentazione

nel caso di accesso ai campi della struct tramite puntatore si può utilizzare l'operatore '->' suffisso all'espressione indirizzo

nel caso si conosca il nome della variabile si tipo struct si può accedere ai campi con l'operatore '.' suffisso al nome della variabile

Tutta questa diventa una variabile, di cui noi possiamo conoscere o il nome, oppure avere un pointer che ci indirizza a dove questa variabile esiste in memoria.

Sia il nome e sia il pointer ci possono permettere di accedere ai campi della struttura.

Un esempio

```
#include <stdio.h>

typedef struct _table_entry{
    int x;
    float y;
} table_entry;

table_entry t;
table_entry *p = &t;

void main(void){
    printf("please provide me with an INT and a FLOAT\n");
    scanf("%d %f",&(t.x),&(t.y));
    // scanf("%d %f",&(p->x),&(p->y)); //equivalent to previous line
    // scanf("%d %f",&((&t)->x),&((&t)->y)); //again equivalent
    printf("x is %d - y is %f\n",t.x,t.y);
}
```

