Lezione 5 - 09 03 18

Friday, March 9, 2018 8:37

Se una variabile è dichiarata fuori dal main ha visibilità globale.

↳ finiscono in un'area di memoria chiamata statica

↓ determina in modo statico la posizione

Vengono anche inizializzate a $\emptyset$.

Le variabili dichiarate nel main vengono inserite in uno stack

+ uno stack (contesto) per ogni chiamata a funzione (activation record)

L'insieme di questi stack è detto stack di programma.

Le variabili locali non sono inizializzate a $\emptyset$.

```
int main () {
    f1();
    f3();
}
```

→ f3 scrive sopra lo stack di f1 che ha finito, può trovare cose sue

Blocco in C: attraverso le graffe, crea un activation record per memorizzare variabili temporanee.

È presente anche uno heap che cresce in senso contrario allo stack

Vi si accede attraverso la malloc: viene chiamata dinamicamente.

```
char * p;
p = (char *) malloc (12);
```

alloca 12 byte

↳ returna un puntatore a void, necessita un cast

Man mano che alloco mi avvicino allo stack.

Però è possibile deallocare l'area di memoria attraverso la free

```
free (p);
```

Quando l'heap è finito, la malloc restituisce NULL (puntatore zero)

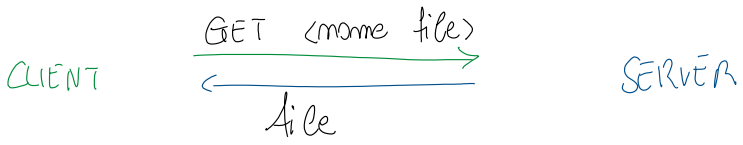
Se si prova ad usare il puntatore a NULL, il sistema operativo dà un errore di segmentation fault.

7. APPLICATION:

Ortogonalità dei livelli: vedere come se comunicassero con il Cop relativo, senza preoccuparsi di cosa succede sotto.

Client/Server → web

Il primo protocollo web era HTTP 0.9

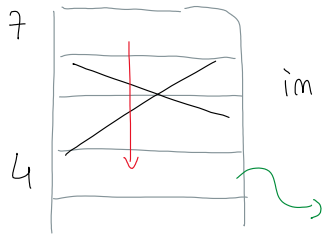


URL = Uniform Resource Locations

↳ In che macchina e dove si trova il file richiesto.

HTTP 0.9 → HTTP 1.0 → HTTP 1.1 → HTTP 2.0

Il protocollo si è evoluto nel tempo.



in HTTP 0.9 non erano stati considerati i livelli 6 e 5.

Service access point
SAP 4

Il livello 4 ci permette di scambiare flussi di byte in maniera simile alla scrittura di file.

open è una chiamata a sistema

↳ Restituisce un int

fopen è una funzione C che al suo interno ha anche una open

↳ Restituisce un puntatore a STRUTTURA

Socket è una chiamata a sistema che restituisce un intero come file descriptor

(simile a open) → deve pensare ad una rete

```c
#include <stdio.h>

int main (){

    FILE * f;
    f = fopen("prova.txt","wt");
    if(f == NULL){
        printf("Errore apertura file\n");
        return 0;

    }

    fprintf(f,"Ciao\n");
    fclose(f);
```

(*)

```
#define NORD 0
#define SUD 1
#define EST 2
#define OVEST 3

int n;

n = NORD;
```

DEFINE sono comandi all'editor del preprocessore => viene manipolato il sorgente

=> Standardizzare un concetto

(*) in stdio.h c'è #define NULL 0 , potevo scrivere == 0

MACRO: (non è una chiamata a funzione, ma una regola di sostituzione testuale)

```
# define MAX ( x, y ) ( x>y ) ? x : y

int m1, m2;

m1 = 1;
m2 = 2;

printf ("%d \n" , MAX (m1,m2) );
```