SETI 26/9

I programmi sono eseguiti dall'hardware della CPU
La CPU esegue: fetch, decode e execute

- **Fetch** : estrae dalla RAM la prossima informazione da eseguire
- Decodifica **(decode)** le informazioni lette poi eseguo **(execute)**

Tutte le CPU hanno un registro, che indica l'indirizzo dove eseguire

Come facciamo a sapere se succede qualcosa nel mondo esterno??
Interact : segnale di controllo da parte delle periferiche che comunica che l'operazione è stata eseguita ecc

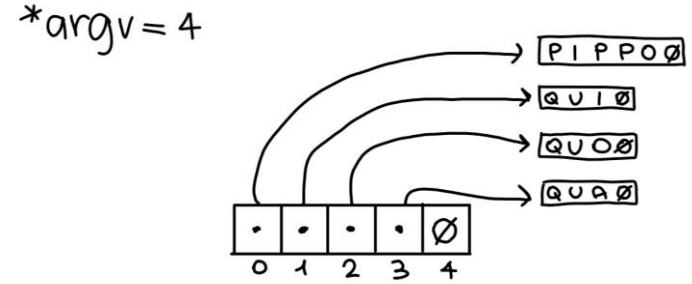
Sistema operativo : gestisce l'hardware e le periferiche
Programma (concetto statico) -> **processo (programma in esecuzione)**
Un programma è un insieme di istruzioni
Il sistema operativo rende isolati i processi
Spazio di indirizzamento: virtualizzazione della RAM, il programma ha l'illusione (creata dal sistema operativo) di avere tutta la memoria possibile
Nucleo, **kernel** del sistema operativo: è lo strato più basso, gestisce l'hardware e virtualizza le risorse come CPU, memoria, ecc
Ogni hardware ha delle regole diverse, il sistema operativo capisce come "usarlo" con i driver -> I **driver** sono pezzi di software che si inseriscono nel sistema per imparare ad "usare" una periferica; sono necessari ma sono anche una minaccia per la sicurezza. Se c'è una vulnerabilità nel driver può intaccare tutto il sistema, perché hanno gli stessi privilegi di un sistema operativo.

```c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

int num;

int main(int argc, char *argv[])
{
        if (argc != 2) {
                fprintf(stderr, "usage: %s <value>\n", *argv);
                return EXIT_FAILURE;
        }
        num = atoi(argv[1]);
        for(;;++num) {
                printf("(pid:%jd) &num=%p num=%d\n", (intmax_t)getpid(), (void*)&num, num);
                sleep(1);
        }
}
```

argc : numero di argomenti che arriva dalla riga di comando
atoi: ask to int, converte da stringa a intero (stoi in c++)
pid: process identifier, un intero maggiore o uguale a 1 che identifica il processo
%p : stampa il puntatore
char *argv[ ] : un array di puntatori



Modalità utente e modalità privilegiata : per isolare e proteggere i processi

- se un processo potesse parlare con l'unita disco (esempio) potrebbe leggere tutto il suo contenuto, compreso i file degli altri utenti ecc
- la modalità privilegiata può anche essere chiamata kernel, perché parla con l'hardware ecc
- la modalità non privilegiata, ossia utente, usata per far girare i processi utente

esistono istruzioni per passare da modalità utente a privilegiata, ma in modo controllato, ossia **chiamata di sistema (system call)**, dove l'utente può richiedere al sistema operativo di fare qualcosa (esempio: leggere un file)
il sistema operativo, in particolare il kernel è una parte del software efficiente ma delicata, deve essere sicura, non deve avere bug (o comunque fixxarlo appena trovato)

unix è un sistema operativo nato metà degli anni '60, semplice e potente. ha ispirato vari unix-like open source : linux ecc