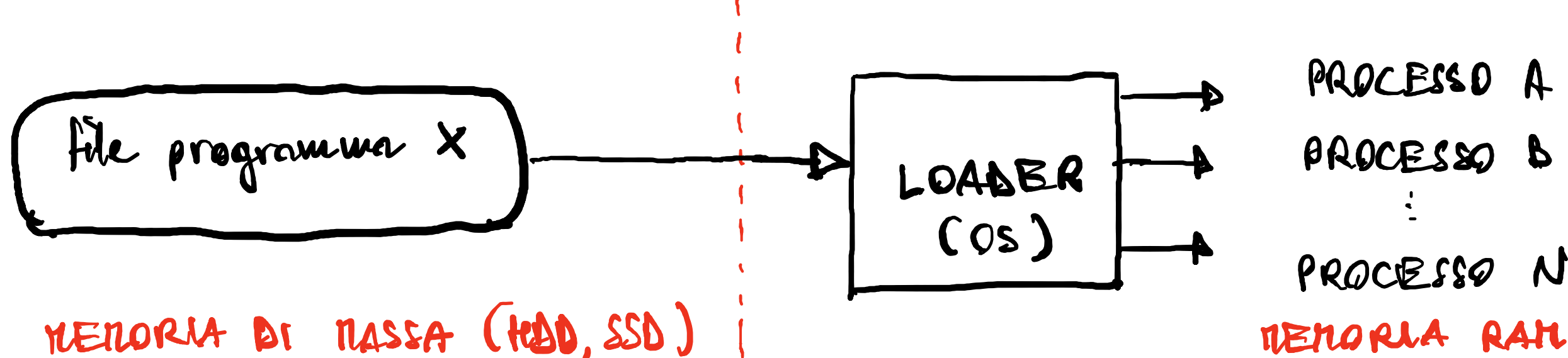


ESECUZIONE DEI PROGRAMMI

Un programma in esecuzione prende il nome di **PROCESSO**
 Un processo è un'istanza di programma in esecuzione

Più istanze dello stesso programma in esecuzione concorrente sono rappresentate da **processi distinti**

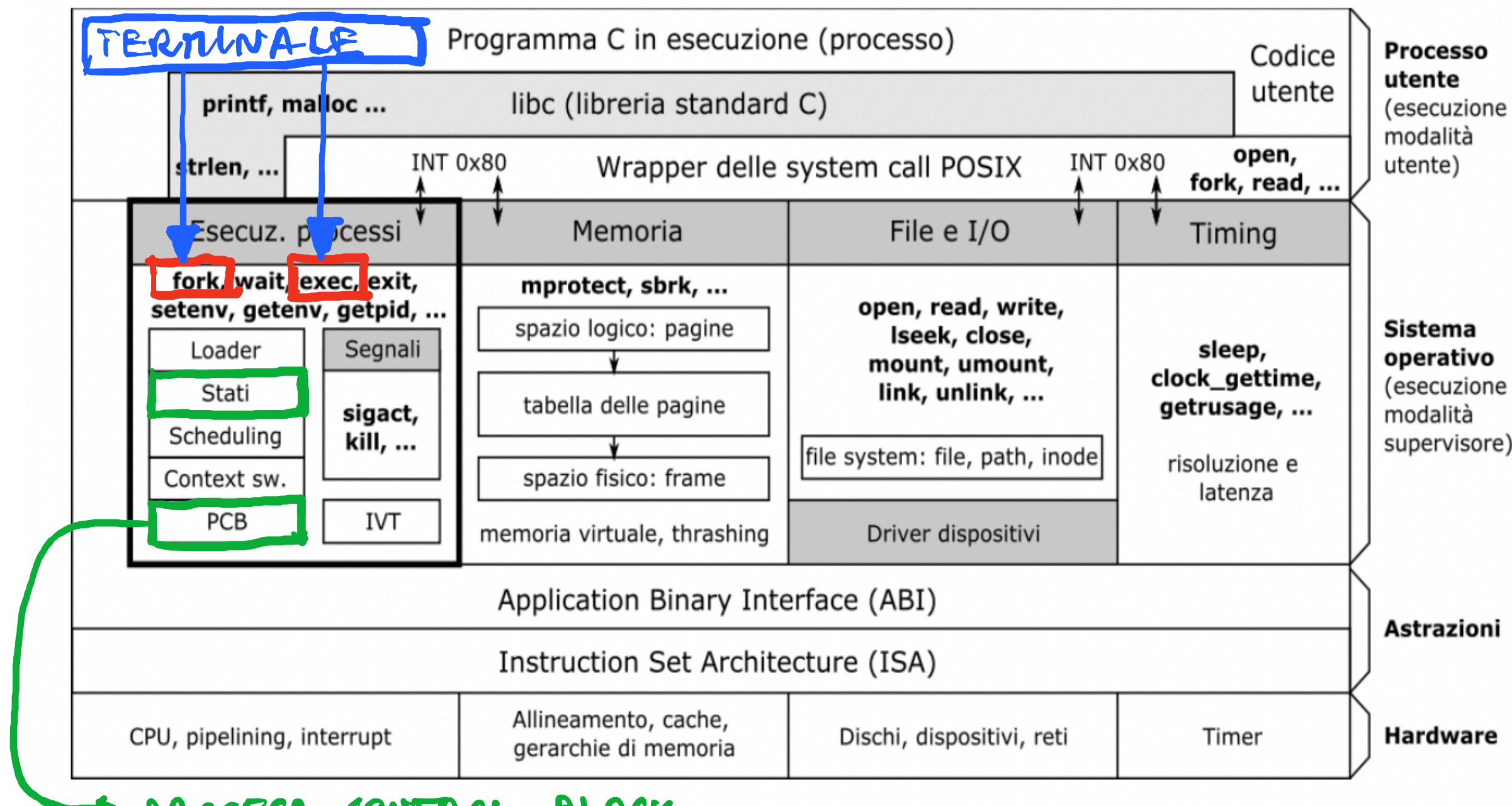
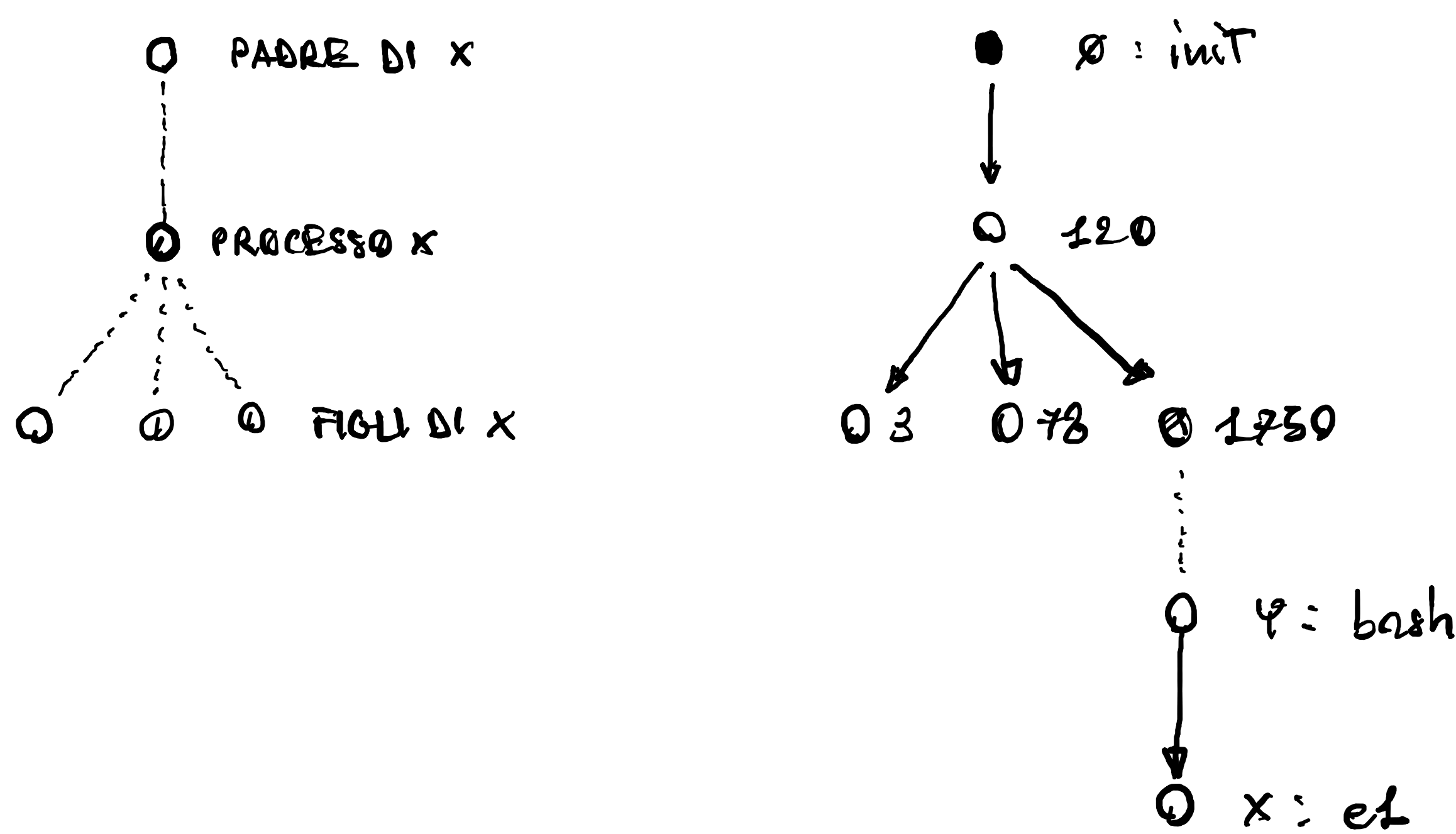


L'esecuzione di ogni processo è **INDIPENDENTE** da quella degli altri processi (a meno che il programmatore non abbia previsto altrimenti)

Un processo nel sistema è identificato univocamente da un numero detto **PROCESS IDENTIFIER (PID)**

Due processi **CONCORRENTI** hanno **SEMPRE PID DISTINTI**

I processi in esecuzione all'interno del sistema sono organizzati in una struttura ad albero



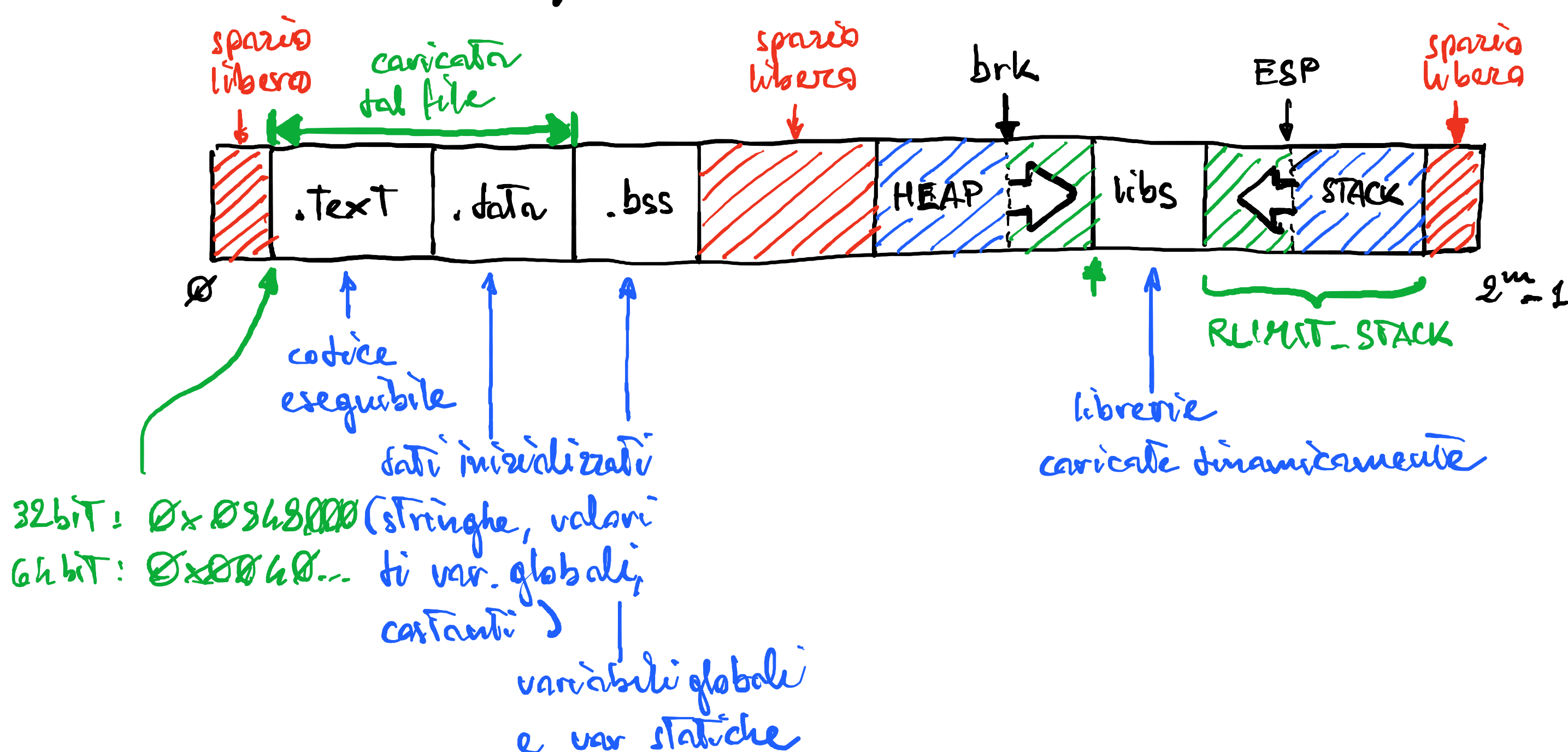
Un processo è costituito dai seguenti elementi:

- La sua **IMMAGINE DI MEMORIA** che contiene:
 - Codice
 - Dati definiti dal programmatore
 - Spazio per le variabili
 - ...
- Lo **STATO DELLA CPU** (registri, ...)
- Le **RISORSE IN USO** (es.: file aperti, connessioni di rete, ...)
- METADATI** (es.: PID, user, ...)

I processi per lavorare nella loro immaginetta memoria **NON** accedono direttamente alla memoria fisica (RAM), ma vi accedono attraverso una **astrazione (memoria virtuale)** gestita dal sistema operativo.



Come è fatta l'immagine di memoria di un processo:



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

int i[10000000], j = 10;

int main(int argc, char* argv[]) {
    static char *k = "hello";
    int *p = malloc(sizeof(int));

    free(p);
    printf("&i=%p | &j=%p | &main=%p | &argc=%p |",
        &i, &j, &main, &argc);
    printf("&k=%p | k=%p | &p=%p | p=%p\n", &k, k,
        &p, p);
    pause();
    return 0;
}
```