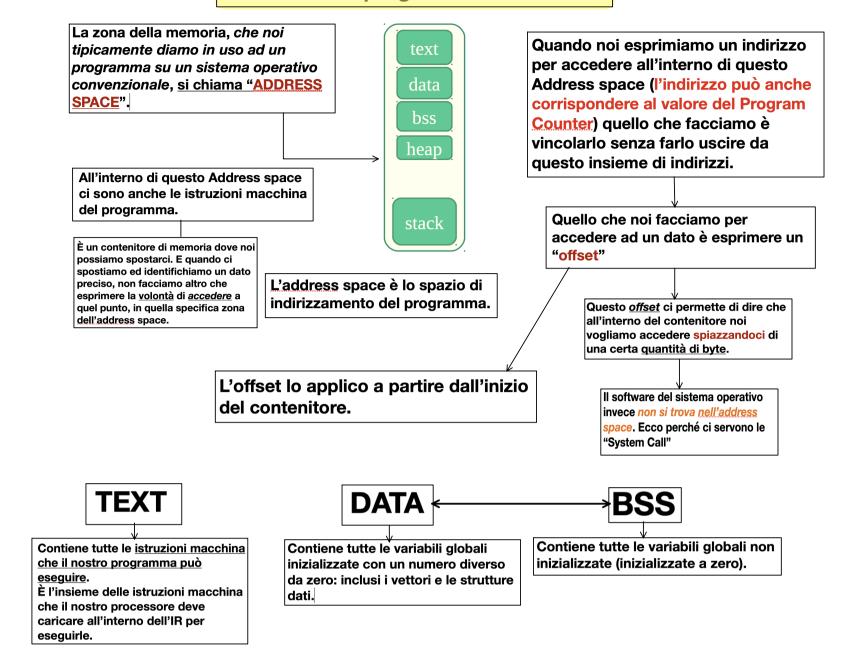
mercoledì 15 marzo 2023 16:42

Anatomia dei programmi in memoria





È la zona che ci permette di chiedere al sistema operativo l'attività che vogliamo (ovviamente ci serve una System call) dove però noi possiamo utilizzare memoria che originariamente non utilizzavamo. Quindi quando andiamo ad allocare più memoria del dovuto tramite una funzione di malloc() in C, interagiamo con questa area che si chiama heap.

STACK

Abbiamo sicuramente una stack area, ossia una zona che utilizziamo per le operazioni che riguardano il mantenimento del Program Counter per ripassare il controllo nel momento in cui abbiamo una variazione di controllo del flusso d'esecuzione, ma abbiamo anche il mantenimento delle variabili locali delle varie funzioni che andiamo a chiamare.

SIA L'heepe le STACK CRESCOND O devessione devante la vita ell programmer;

PER EVITARE (HE QUESTE DUE AREE SI SCONTRIND Puello du succede e che los sraces

quandos e vuoro parte dall'indirizzo più altro e quando si riempie un versor

L'indirizzo più BASSO, quindi cresce dall'altro verso ir BASSO, il nio heap invece

CRESCERO DAL RASSO VERSO L'altro;

SE le due avoir si scontrano ir programma ha finito la MEMORIA.

Una variabile puntatore contiene un indirizzo di memoria che fa riferimento ad un'altra locazione nell'address space. Fisicamente il puntatore si comporta come una variabile normale, può finire nel campo bss o in data, ma comunque ha un suo indirizzo e punta ad una locazione determinata.

Identifichiamo un'altra posizione dell'address space tramite i puntatori