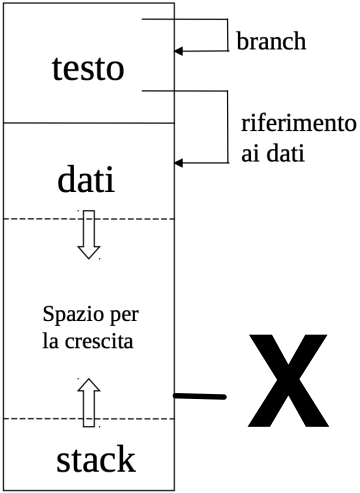


Immagine di memoria e riferimenti

L'address space si trova all'interno della memoria RAM.



Quando un THREAD in esercizio si trova ad eseguire un'istruzione, per esempio, in un determinato punto X, e vuole cercare di accedere a delle informazioni che sono all'interno della memoria, questa istruzione deve arrivare a toccare oggetti che sono all'interno della RAM. Però attenzione: l'istruzione in quel punto vuole andare a toccare qualcosa che è all'interno del contenitore dell'address space. Quindi nel momento in cui questa istruzione è in esercizio chiaramente noi abbiamo il problema di determinare il riferimento che questa istruzione sta esprimendo.

Per esempio se ho un'istruzione all'interno della zona testo dell'address space e il thread l'ha appena pescata per andare in esecuzione, magari abbiamo un riferimento ad un'altra istruzione a cui dobbiamo saltare. Oppure magari l'istruzione è un accesso esplicito a dati che sono all'interno dell'address space, e quindi viene espresso un punto dell'address space che esprime dove il dato a cui siamo interessati è presente. Ma tutto questo deve essere sintetizzato come un accesso effettivo alla RAM, perché siamo in RAM. Il processore quando normalmente porta avanti il semplice fetch delle istruzioni di una certa applicazione, in realtà sta esprimendo delle posizioni delle istruzioni all'interno di questo address space, ma tutto questo alla fine deve essere trasformato come un riferimento alla RAM.

Il testo contiene riferimenti a indirizzi di memoria associati a

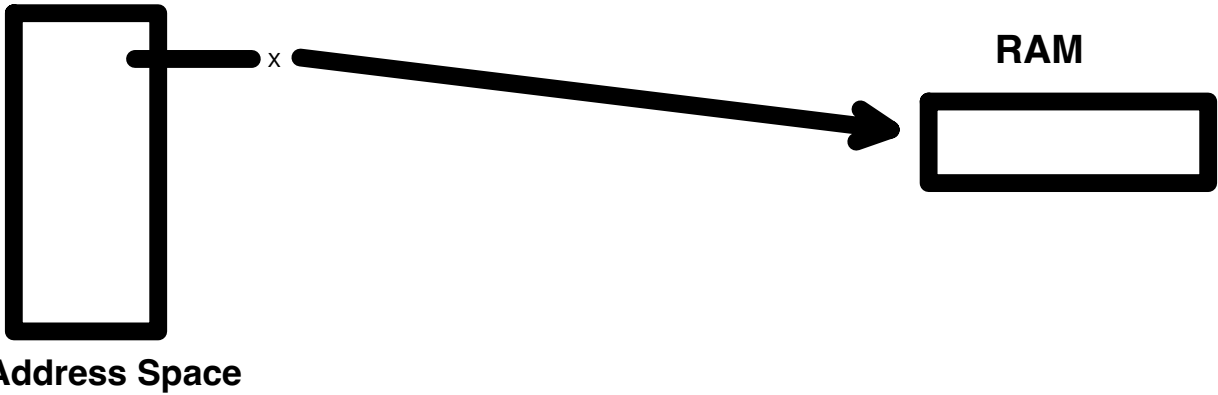
1. istruzioni (salti condizionati e non, chiamate di subroutine)
2. dati (indirizzamento assoluto etc.)

L'associazione di istruzioni e dati agli indirizzi di memoria, può essere fatta in maniera differente all'interno dei sistemi operativi. Questo si chiama In informatica il binding è il processo tramite cui viene effettuato il collegamento fra una entità di un software ed il suo corrispettivo valore. Nei termini della programmazione a oggetti, la decisione circa l'attributo o il metodo da richiamare in un dato momento dell'esecuzione del programma viene effettuata grazie al binding.

"Quando parliamo di BINDING indichiamo la possibilità di determinare la posizione di una determinata istruzione che vogliamo eseguire e del rispettivo salto una volta che eseguiamo la prima, oppure del dato che vogliamo toccare in un'istruzione che stiamo eseguendo."
- Prof. Francesco Quaglia 29-04-2021.

A seconda del tipo di binding che abbiamo all'interno del nostro sistema operativo, abbiamo delle dipendenze e dei condizionamenti su come possiamo assegnare la memoria di lavoro ai processi.
Cosa significa?

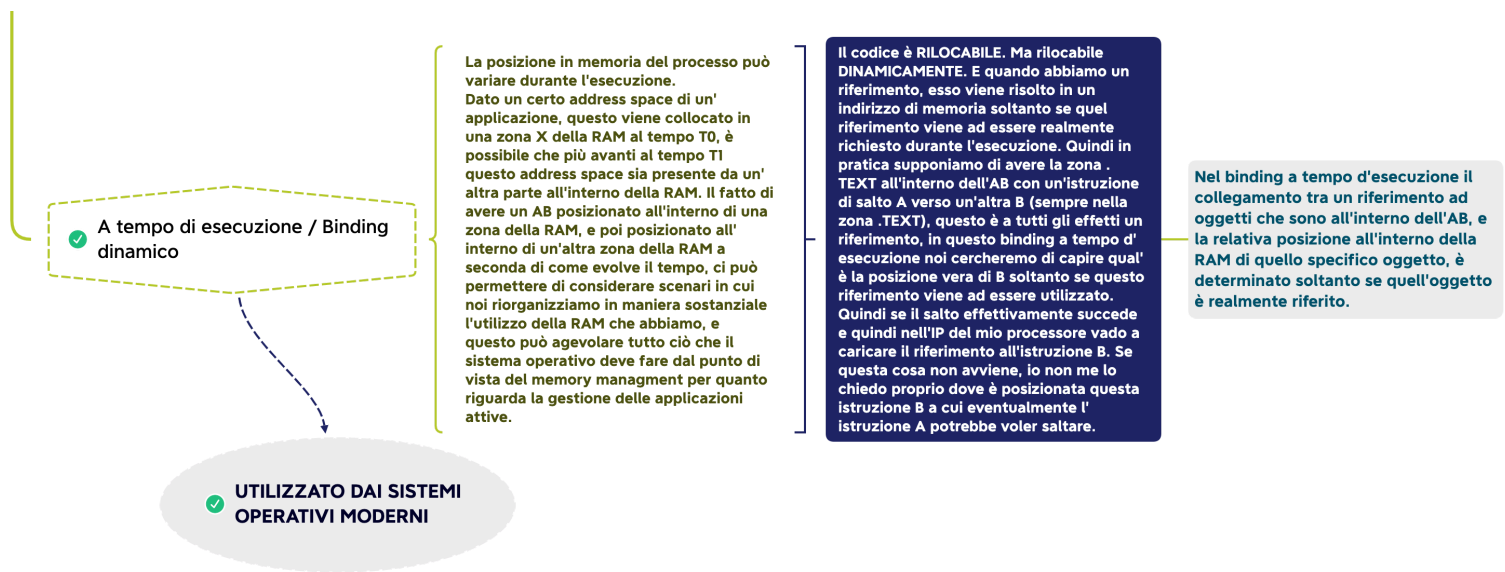
Ho l'address space di un'applicazione che io devo mandare in setup perché questa applicazione X è stata richiesta per l'esecuzione, devo andare a segnare all'interno della RAM, la posizione di questo oggetto ADDRESS SPACE. Io poi vorrei utilizzare un certo punto di questo address space, perché in pratica sto facendo il fetch di quella istruzione allora vado ad identificare la zona della RAM in cui quella istruzione è presente, e a seconda di come questa cosa viene fatta ovviamente questo PONE un vincolo su come noi collochiamo questo address space all'interno della RAM.



Address Space

Sicuramente all'interno dei sistemi abbiamo tecniche diverse per identificare qual è la posizione effettiva all'interno della RAM delle informazioni che stiamo utilizzando, le tre tecniche di base sono le seguenti:





ESEMPI DEL BINDING!

