

Gestione di pagine logiche in Windows

P43 5-5-21
01:14:57

Come facciamo in windows, all'interno di un address space, ad allocare pagine virtuali? **Allocare pagine virtuali significa esattamente mappare queste pagine all'interno dell'address space.**



Abbiamo un API di sistema che ci permette di fare questa cosa: VirtualAlloc.

Essa ci permette di specificare in che punto dell'address space (indirizzo) devo allocare la pagina, poi specificare una certa quantità di byte da quel punto in poi, e infine abbiamo la possibilità di passare dei parametri che ci indicano la location type e il tipo di protezione.

Il tipo di protezione significa che possiamo andare a lavorare lì in scrittura/lettura/esecuzione, ma con la location type possiamo specificare numerose possibilità:

MEM_COMMIT MEM_RESERVE e MEM_RESET.

Se chiamiamo una VirtualAlloc e come terzo parametro passiamo MEM_COMMIT indichiamo che la memoria che stiamo allocando virtualmente (mmappando) è materializzabile. Quindi chiediamo che queste pagine ci siano all'interno del contenitore e poi le utilizziamo anche, e quando le usiamo queste pagine vengono ad essere materializzate in RAM.

Se chiamiamo una VirtualAlloc e come terzo parametro passiamo MEM_RESERVE indichiamo che le pagine che stiamo mmappando non sono materializzabili. Stiamo cambiando la struttura dell'address space soltanto quello che riguarda il collocamento delle pagine ma non stiamo rendendo realmente quelle pagine riutilizzabili.

Se chiamiamo una VirtualAlloc e come terzo parametro passiamo MEM_RESET indichiamo che una serie di pagine che eventualmente potevano anche essere state materializzate sono tali per cui il contenuto che è attualmente materializzato per quelle pagine non ci interessa più.

Quindi la pagina dall'address space poteva anche essere materializzata in RAM, ma questa pagina è tale per cui il suo contenuto originale non ci interessa più, quindi non è più vero che nella page table dobbiamo mantenere l'associazione tra la pagina nell'address space e il suo corrispondente frame fisico.

Questo vuol dire che se riaccendiamo a questa pagina essa verrà materializzata da qualche altra parte in RAM. È una pagina nuovamente anonima.

Abbiamo la system call VirtualFree che ci dice che dato un indirizzo per una certa quantità di memoria bisogna applicare un tipo di gestione awfreelype che può essere:

- MEM_DECOMMIT: Se quella pagina è materializzata la dematerializziamo ma quelle pagine nell'address space restano comunque mappate.
- MEM_RELEASE: Rilasciamo la specifica regione di pagine, ossia equivalente a Memory-UNMAP che noi abbiamo su UNIX.

Per cambiare i permessi abbiamo la system call VirtualProtect in cui dato un indirizzo di una certa quantità di byte (quindi di una certa zona di pagine) applicare una nuova protezione specificabili con delle macro quali PAGE_READONLY PAGE_READWRITE PAGE_EXECUTE, e l'ultimo parametro è un pointer ad un'area in cui il sistema ci riporta qual era la vecchia protezione che stiamo alterando per effetto di questa chiamata.