

Il thrashing

Il thrashing si ottiene in un sistema di memoria virtuale quando aumentiamo troppo il livello di multiprogrammazione. Quindi invece di favorire l'uso del processore - perché abbiamo un bel blocco di applicazioni che sono parzialmente in memoria e quindi possono essere ready per prendere il controllo della CPU - e invece l'uso del processore diminuisce. Queste applicazioni sono troppe: se ne abbiamo troppe tipicamente tendiamo a creare dei resident sets troppo piccoli (quindi troppa poca memoria fisica per ciascuna di queste applicazioni). Quindi se noi aumentiamo il livello di multiprogrammazione arriviamo ad un massimo nel livello di uso del processore, ma oltre decresce l'uso della CPU. Questo perché tutte queste applicazioni quando vengono schedate in CPU generano subito page fault. Page fault significa che dobbiamo andare bloccati in I/O ad aspettare che arrivino delle cose che vengono ricaricate in memoria per effetto della sostituzione. Quindi non è più vero che aumentando il livello di multiprogrammazione possiamo mantenere la probabilità di utilizzare il processore in maniera adeguata evitando di avere che le applicazioni siano tutte in blocco, no le applicazioni vanno in blocco perché c'è esattamente il sistema di memoria virtuale, e ci sono dei resident sets troppo piccoli che inducono frequenze di page fault eccessive e quindi blocchi dei thread che devono attendere che le pagine vengano ricaricate a partire dalla swap area.

- tipicamente un aumento del livello di multiprogrammazione dovrebbe favorire l'uso del processore e dei dispositivi
- ma un grado di multiprogrammazione eccessivo tende a creare resident sets troppo piccoli e l'uso del processore diminuisce (la frequenza di page fault diviene eccessiva per tutti i processi attivi i quali rimangono bloccati in attesa di caricamento delle loro pagine)

