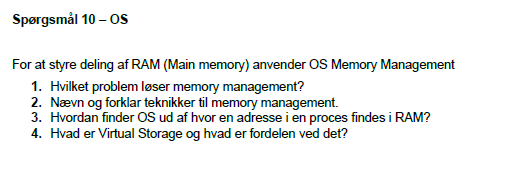
Spørgsmål 10



At processor bliver udført effektivt hurtigt og med så lidt spild som muligt, og at kunne skifte mellem processor når det er “bedst”.

Operativsystem skal holde øje med hvor meget fri data der er, og udgive det til ressourcer så de kan udføres i en rækkefølge der er mest effektiv. Krav til Memory Management er relocation, protection og sharing.

Der er 2 hovedpunkter. Der er fixed partitioning og dynamic partitioning. Dette var sådan det startede , fixed partitioning blev ressourcer placeret efter og var tildelt en bestemt mængde lige meget hvor meget ressourcen brugte. Det betød der var ofte meget internal fragmentation. Dynamic partitioning blev ressourcerne tilføjet med det de faktisk brugte af ressour. Desværre resulterede dette ofte i meget external fragmentering.

Senere hen kom Simple paging hvilket var en videreudvikling af fixed partitioning. Det virkede som vidt ganske meget ens som fixed partitioning, i steder for blev ressourcerne opdelt i meget små ens bider, dette resulterer i meget mindre internal fragmentation, dog så var det der stadig. Dette betød også at hele ressourcen ikke længere var tvunget til at være inde i RAM, og dele kunne tilføjes, dette betød også at man ikke behøvede at havde alle dele lige efter hinanden da man holdte styr på hvor en del af ressourcen sluttede og den næste startede. Derudover kom der selvfølgelig og simple segmentering, hvilket virker meget ligesom simple paging, i og ved den er opdelt i små bidder, men ligesom dynamic partitioning, så er den ikke tvunget ind i en bestemt størrelse, da delene er meget mindre så er der også mindre external fragmentation. Ligesom simple paging. holdes der styr på hvornår delene af ressourcen starter og slutter samt deres størrelse, og at ikke hele ressourcen skal være inden i RAM.

RAM finder position via de relative/logiske adresse som så omskrives til den fysiske adresse, i simple segmentation. Den logiske adresse er opdelt i 16 bits. 4 bits er til for at vide dens nummer, og 16 bits er dens offset hvilket er længde + basen.

Virtual storage er hvor dele af ressourcen er gemt inden i harddisken og kun bliver taget op når den skal bruges. Dette gør at der er plads til flere ressourcer inden I RAM og der kan arbejdes meget mere effektivt, her tager man brug af flere politikker for kunne udfører arbejdet.

Fetch, replacement og cleaning.

* Fetch er hvor en page bliver hentet ind i RAM, her kan du enten kører en “Demand paging” hvor page bliver hentet når den skal bruges, eller Prepaging hvor der bliver hentet flere pages ind af gangen.
* Replacement er hvor der fjernes en page for at en ny en kan komme ind, der er 3 forskellige algoritmer der almidelige at bruger. “First In First Out”, hvor den første page der kom ind er den første der fjernes når en ny page skal ind. “Least Recently Used” hvilket er hvor den page der sidst er brugt bliver fjernet for en ny. og til sidst “Clock” hvilket bygger på de ovenstående 2, hvor du har en tæller der tæller op hvis en page bliver brugt. Og en page fjernes som den første der kom ind af dem der er brugt mindst.