

Laboration 4

Paging

Henrik Bäck
850611-6253
Mathias Andersson
850424-6292

Operating Systems DAVB01
VT 2005

Handledare: **Nils Dåverhög**
Hans Hedbom
Thijs Jan Holleboom

Inlämnad **2007-02-12**

Innehållsförteckning

1. ANTAGANDEN	4
2. ÖVERSIKT.....	4
3. DETALJERAD BESKRIVNING.....	4
4. SLUTSATS	4
5. PROBLEM.....	4
6. SAMMANFATTNING.....	5
7. REFERENSLISTA.....	5

1. Antaganden

Det antogs att FIFO skulle tillämpas vid användning av det fysiska minnet. Vidare antogs att information om filens längd var känd innan exekvering av programmet skedde.

Gällande sidtabellen antogs det att den skulle bestå av en array av struct innehållande två variabler av typer *integer* (för frame) och *boolean* (för valid bit).

2. Översikt

Programmet simulerar sidhantering och läser in data från fil, mellanlagrar datan i minnet och skriver sedan önskad data till skärmen. På detta sätt behöver inte data från fil läsa så länge den finns i minnet.

3. Detaljerad beskrivning

Programmet består av två stycken viktiga arrays samt en matris som representerar det fysiska minnet. Denna matris har plats för så många ord som är fördefinierat enligt `fileSize`. En array representerar vilka sidor i det fysiska minnet som är lediga respektive upptagna. Den andra arrayen består av structs. Dessa innehåller information om vilken frame som den aktuella sidan ligger i samt om sidan finns inläst i minnet eller inte.

När en file exekveras genom programmet kommer den att iterera genom hela filen och läsa en sida i taget och lägga det i det fysiska minnet. Därifrån kommer sedan informationen att skrivas till skärmen. Så länge det ord som skall skrivas finns i minnet behövs ingen läsning från filen göras. Genom att kontrollera om den frame som skall läsas är giltig i sidtabellen kan man avgöra detta.

Det finns möjlighet att läsa in valfritt ord från hela filen. Genom att ange positionen på det ord som skall läsas kan man avgöra vilken sida och vilken offset inom denna sida som skall skrivas till skärmen. Om sidan finns i minnet är det bara att hämta ordet från korrekt position annars måste en läsning från fil göras.

Det finns en möjlighet att skriva ut den aktuella sidtabellen till skärm.

4. Slutsats

Genom att läsa en fil på detta sätt reduceras antalet läsningar från fil. Eftersom mycket av den datan som skall läsas finns i minnet behövs den endast hämtas därifrån.

5. Problem

Från början var det fysiska minnet deklarerat som

```
ord physicMem[page][memSize]
```

Det uppstod flertalet underliga fel då information skrevs till matrisen.

Programmet skrev data till andra minnesutrymmen än vad som var avsett.

Detta berodde på att antalet minnesplatser var för få. Genom att ha förväxlat [page] och [memSize] vid deklarationen fick vi endast 4 platser á memSize stora. Problemet löstes genom att byta ut deklarationen mot

```
ord physicMem[memSize][page]
```

6. Sammanfattning

Laborationen har gett fördjupad förståelse i hur paging i ett operativsystem kan gå till. Laborationen löstes relativt snabbt efter att förståelse för **vad** och **hur** labben skulle lösas erhöles.

7. Referenslista

Manual (*man*) för kommandon *fread*, *fsead* och *fopen*. Denna manual fanns i RedHat Linux 9.

