

# **Laboration 2**

## **Semaforer**

**Henrik Bäck**  
**850611-6253**  
**Mathias Andersson**  
**850424-6292**

**Operating Systems DAVB01**  
**VT 2005**

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

Inlämnad **2005-04-19**

HENRIK BÄCK  
MATHIAS ANDERSSON  
Datavetenskap

2005-04-19  
Laboration 2  
Semaforer

2(5)  
Operating Systems 5p  
DAVB01

## **Innehållsförteckning**

<b>1. ANTAGANDEN .....</b>	<b>4</b>
<b>2. ÖVERSIKT.....</b>	<b>4</b>
<b>3. DETALJERAD BESKRIVNING.....</b>	<b>4</b>
Programmet main.....	4
Programmen prc_a, prc_b och prc_c .....	4
<b>KÖRNING AV MAIN OCH DESS DOTTERPROCESSER ...</b>	<b>4</b>
<b>4. PROBLEM.....</b>	<b>5</b>
<b>5. SAMMANFATTNING.....</b>	<b>5</b>
<b>6. REFERENSLISTA.....</b>	<b>5</b>
<b>7. BILAGOR.....</b>	<b>5</b>

## 1. Antaganden

Det antogs att semaforerna skulle representeras av filer i den aktuella arbetskatalogen.

## 2. Översikt

Ett huvudprogram startar upp tre stycken dotterprocesser. Dessa dotterprocesser har var och en till uppgift att skriva sin bokstav till skärmen på given signal. De tre programmen skriver A, B respektive C. En dotterprocess får bara skriva till skärmen när denna får en signal att göra så. Totalt skall man genom att starta *main* kunna få följande skrivet till skärmen: ABC AB AC AB AC AB AC AB ...

Detta pågår tills användaren trycker CTRL+C. När detta sker avbryts programmet och alla dotterprocesser samt signaleringsfiler tas bort.

## 3. Detaljerad beskrivning

### ***Programmet main***

Programmet *main* består av en funktion som rensar systemet från de processer den skapat samt tar bort de filer som används för att signalera till dotterprocesser. Main-funktionen i programmet startar upp tre processer där var och en av processerna har en specifik uppgift. Därefter väntar programmet på att användaren trycker CTRL+C för att köra funktionen för rensning och sedan avsluta programmet.

### ***Programmen prc\_a, prc\_b och prc\_c***

Dessa tre program utför i stort sett samma uppgift med den stora skillnaden att *prc\_a* skriver ett A till skärmen, *prc\_b* ett B och *prc\_c* ett C. Det andra som skiljer dem åt är vilken av de andra två processerna som skall signaleras efter utförd körning. Programmen loopar i en oändlig loop och inväntar sin signal för att få köras.

### **Körning av main och dess dotterprocesser**

Vid körning av programmet *main* startas i tur och ordning processerna *prc\_a*, *prc\_b* och *prc\_c* upp. När barnprocesserna startats av *main* kommer *prc\_a* att skriva A. Efter detta ger den klartecken till *prc\_b* att skriva B vilket denna gör och signalerar härpå *prc\_c* att skriva C. Efter detta kommer *prc\_c* att signalera till *prc\_a* att det är dennes tur att använda skärmen. *Prc\_a* skriver A till skärmen och signalerar därefter antingen *prc\_b* eller *prc\_c*. Detta beror helt och hållet på vilken av dessa processer som senast blev signalerad. Programmet fortsätter att loopa ut AB och AC om vart

annat tills användaren avbryter det hela genom att trycka CTRL+C. Då avbryts utskriften till skärmen och programmet avslutas.

#### **4. Problem**

När operationerna *open* och *creat* skulle användas blev det problem med vilka flaggor som skulle användas. Systemet rapporterade (-1) som returvärde vilket resulterade i att processerna inte kunde skapa sina filer och avslutades. Genom att läsa igenom manualen för *open* igen förstods senare vilken av flaggorna som kunde användas tillsammans med vilket kommando.

#### **5. Sammanfattning**

Laborationen kunde enkelt lösas efter att problemet med *open* och *creat* hade lösts. Efter detta flöt laborationen på bra och löstes inom loppet av någon timma. Körning av programmen gick som förväntat och gav korrekt resultat.

#### **6. Referenslista**

Manual (*man*) för kommandon *open* och *creat*. Denna manual fanns i RedHat Linux 9.

#### **7. Bilagor**

**Bilaga A** – Programkod för moder och barnprocesser.