OPERATIVSYSTEM 2008, RÄKNEÖVNING 2, v. 39, deadline 29.9.2008

OBS Uppgift 5 måste besvaras!

1. Följande program är givet:

```
/* osro_2_1.c */
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

main() {

  FILE *fp;
  pid_t pid;

  fp = fopen("testfile", "w+");
  pid = fork();
  if (pid) {
    fprintf(fp, "Parent writing to file\n");
    fclose(fp);
  } else {
    fprintf(fp, "Child writing to file\n");
    fclose(fp);
  }
}
```

   Provkör programmet, vad händer? Hur kan det komma sig att båda processerna skriver till samma fil? Processer har ju inte delade filer? Hur är det möjligt att stänga en fil två gånger? (2p)

2. Undersök ”De ätande filosoferna” och lösningen på sidan 127 i boken (dvs. Tanenbaum), alternativt sidan 31 i föreläsningsfolierna. Vad händer om man i rutinen *put_forks* sätter status till THINKING **efter** de två anropen till test, istället för **före**? Ge ett illustrativt exempel (2p)

3. Följande given en resursallokeringstabell för ett system där det finns 5 resurser av typ 1 (R1) samt 6 resurser av typ 2 (R2). Är systemet i ett säkert tillstånd? Visa varför/varför inte! (2p)

Process	Max behov R1	Max behov R2	Allokerat R1	Allokerat R2
P1	3	3	1	1
P2	2	3	1	1
P3	5	6	1	3

4. *Readers / writers*-problemet finns beskrivet finns beskrivet på sidan 129 i boken, alternativt sidan 34 i föreläsningsfolierna (IPC). Visa hur man skulle ändra programmet så att *writers* har förtur (dvs. finns det en väntande *writer*, så kan inte flera *readers* använda databasen). (2p)

5. Skriv ett program som använder sig av trådar. Programmet skall producera 3 trådar, som dels visar att de har delat minne, dels visar att lokala variabler (på stacken) inte påverkar varandra mellan trådar. Bifoga programlistning samt testutskrift. (4p)