

OPERATIVSYSTEM 2009, ÖVNING 2, v. 39, deadline 25.11.2010

OBS. Övning inlämnas elektronisk på adressen <https://xprog28.cs.abo.fi/ro.nsf>.

1. Nedan är ett enkelt program för att beräkna en fibonacci-serie (inte det bästa, men det fungerar)

```
/* fibonacci.c */
#include <stdio.h>

int main() {

    long long old[2] = {0,1};
    while (1) {
        long long tmp;
        tmp= old[0]+old[1];
        old[0] = old[1]; old[1] = tmp;

        printf("%li\n", tmp);
    }
}
```

Skriv om programmet så att man skapar en FIFO, dit fibonacci-numret skrivs. Visa att man genom att läsa från FIFO:n nu får fibonacci-serien (se systemkommandot *mkfifo*) (2p)

2. Undersök ”De ätande filosoferna” och lösningen avsnitt 2.5.1 i boken (dvs. Tanenbaum), alternativt sidan 31 i föreläsningsfolierna (forelasning_5.pdf). 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 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)

Ledning: använd Posix threads, tutorial finns på

<https://computing.llnl.gov/tutorials/pthreads/>

och HelloWorld för pthreads (som exempel/start) på

<https://computing.llnl.gov/tutorials/pthreads/samples/hello.c>