

Bankproblem

Vinjett på termin 4 i TDDE47 Processprogrammering och operativsystem

Banken i den lilla byn Snålköping har nyligen beslutat att ta steget in i 2000-talet och investera i ett digitalt system för att hantera alla konton och överföringar. En viktig anledning till beslutet är att invånarna i byn gillar att föra över pengar mellan sina konton, och därmed är det viktigt att det nya systemet kan göra överföringar så effektivt som möjligt. Banken hanterar bara konton och överföringar mellan konton, inte lån och andra liknande tjänster. Alltså ska saldot på alla konton vara positiva.

När banken har testat sitt system ett tag inser de att det inte fungerar som det ska. Exempelvis har anställda noterat att konton ibland fått negativt saldo, vilket absolut inte bör hända. Banken tror att problemet ligger i **transfer**-funktionen, som ska föra över pengar mellan två konton. Koden för funktionen finns nedan. Du kan anta att det bara är **transfer**-funktionen i systemet som använder **accounts**-variabeln. Funktionen kan anropas ifrån flera trådar samtidigt.

```
bool transfer(int amount, struct account *from, struct account *to) {
    bool ok = from->balance >= amount;
    if (ok) {
        from->balance -= amount;
        to->balance += amount;
    }
    return ok;
}
```

Utvecklarna till systemet har föreslagit följande två lösningar på problemet. Dock litat inte banken längre på utvecklarna och har bett er om hjälp att se vilken lösning som är bäst, eller om problemet bör lösas på något helt annat sätt.

```
A:
bool transfer(int amount, struct account *from, struct account *to) {
    lock_acquire(&lock);

    bool ok = from->balance >= amount;
    if (ok) {
        from->balance -= amount;
        to->balance += amount;
    }

    lock_release(&lock);
    return ok;
}
```

```
B:
bool transfer(int amount, struct account *from, struct account *to) {
    lock_acquire(&from->lock);
    lock_acquire(&to->lock);

    bool ok = from->balance >= amount;
    if (ok) {
        from->balance -= amount;
        to->balance += amount;
    }

    lock_release(&from->lock);
    lock_release(&to->lock);
    return ok;
}
```

Se följande länk för detaljer om hur du kan undersöka lösningarna:

<https://www.ida.liu.se/~TDDE47/vinjetter/progvis/>