

TDIU16: Process- och operativsystemprogrammering

Övning på semaforer

Filip Strömbäck

1 Mål

Målet med denna laboration är dels att illustrera hur systemanropen `exec` och `wait` ska fungera i Pintos, och dels att ge övning på hur semaforer fungerar. Denna laboration görs utanför Pintos, vilket gör det lite enklare att se vad som händer (exempelvis med visualisering) och felsöka eventuella problem.

2 Uppgift

I mappen `src/standalone/lab07` finns filen `exec-wait.c`. Filen innehåller bland annat en funktion, `do_work`, som vi tänker oss gör tunga beräkningar. Vi vill därför köra funktionen parallellt i flera trådar för att utnyttja alla kärnor i vår CPU.

För att göra detta innehåller filen funktionerna `exec` och `wait` för att enkelt hantera detta i resten av programmet. Tanken är att funktionen `exec` ska starta en tråd som kör `do_work` i en separat tråd. Man ska sedan kunna vänta på att `do_work` blir klar genom att anropa `wait`. Med hjälp av dessa funktioner kan vi sedan i `main` starta hur många trådar vi vill genom att anropa `exec` flera gånger, och sedan vänta på att alla trådar blir klara med `wait`. Detta görs i funktionen `main` i filen `exec-wait.c`.

Som du säkert märker fungerar inte koden som den ska. Ofta får du att första och andra resultaten är 0, vilket ser fel ut. Detta beror på att koden inte är korrekt synkroniserad. Uppgiften är att lösa de synkroniseringsproblem som finns med hjälp av semaforer.

3 Kompilera och kör koden

Du kan antingen köra programmet i visualiseringsverktyget eller i terminalen. Nedan beskrivs båda sätten.

3.1 Visualiseringsverktyget

Starta visualiseringsverktyget enligt instruktioner på kurshemsidan (välj "Visualiseringsverktyg" i vänstermenyn). På skolans datorer kan du köra kommandot `/courses/TDIU16/progvis/start` men du kan även hämta verktyget till din egen dator om du vill.

När du har startat verktyget kan du sedan välja *File* → *Open...* och sedan välja källkodsfilen för labben. Koden kompileras då automatiskt, och du kan sedan stega fram de olika trådarna i programmet som du vill. Om du ändrat något i källkoden kan du välja *File* → *Reload* (Ctrl+Shift+R) för att se dina ändringar.

Tips: Testa även *Run* → *Look for errors* för att få en indikation om din lösning innehåller synkroniseringsfel eller inte.

3.2 I terminalen

Gå till mappen där `exec-wait.c` finns, och skriv `make exec-wait` för att kompilera koden. Du kan sedan köra programmet med `./exec-wait`. Du kan också använda GDB som vanligt med kommandot `gdb ./exec-wait` eller `gdb -tui ./exec-wait` om det behövs.