

TDIU16: Process- och operativsystemprogrammering

Processhantering i UNIX

Klas Arvidsson, Daniel Thorén, Filip Strömbäck

1 Mål

Den här frivilliga uppgiften syftar till att illustrera hur processer hanteras på UNIX-system, framförallt hur systemanropen `fork`, `exec`, `exit` och `wait` samverkar. Dessa systemanrop kommer du sedan att implementera inuti Pintos med små modifikationer (`fork` och `exec` slås ihop till ett systemanrop i Pintos).

2 Uppgift

Det givna programmet `fork-wait.c` som finns i mappen `src/standalone/labx2/` i Pintos-repositoryt innehåller kod för att starta en ny process på ett UNIX-system (exempelvis Linux) och sedan vänta på att det har kört färdigt.

I det givna programmet skapas en ny process med hjälp av `fork` (man `-s2 fork`). Sedan startar den nya processen ett nytt program med hjälp av `execv` (man `-s2 execv`). I Pintos slås dessa två systemanrop ihop till ett enda som vi kallar `exec`.

Studera koden. Kompilera och kör det givna programmet. Följ instruktionerna som skrivs ut:

```
# Kopiera ett annat program a.out till mappen, exempelvis från tidigare standalone-uppgift.
$ gcc -Wall -Wextra -std=c99 -g fork-wait.c -o fork-wait
$ ./fork-wait ./a.out testing one two three
```

Prova att låta `fork-wait` starta en terminal istället:

```
$ ./fork-wait $(which xterm)
```

Redovisa nu svar på följande frågor:

1. Kör `ps tree $USER` i den terminalen som startades av `fork-wait`. *Vad hände med den andra `fork-wait` som blev barn till den första? (Tips: Jämför processnumren.)*
2. När du kör `fork-wait` exemplet med programmet från föregående uppgift kommer barnprocessen normalt att avsluta innan föräldern (`fork-wait`) ens har börjat vänta. Men trots att barnet redan avslutats och inte finns mer så skrivs summan som returnerades av barnet (`a.out`) ut korrekt av föräldern (`fork-wait`). *Ge förslag på hur operativsystemet (som anropas av `waitpid`) kan tänkas göra så att det fungerar.*
3. När du kör `fork-wait` med en `xterm` så avslutas inte barn-processen (`xterm`) förrän du skriver `exit` i fönstret. Det sker oftast långt efter att föräldern (`fork-wait`) är klar med sitt. Trots det skrivs inte resultatet från barnet ut förrän barnet (`xterm`) avslutats. *Ge förslag på hur operativsystemet (som anropas av `waitpid`) kan tänkas veta när resultatet från angivet processid är tillgängligt.*
4. Kör `fork-wait` med en `xterm`, men avbryt `fork-wait` genom att trycka `Control-C` innan `fork-wait` anropar `waitpid`, och kör sedan kommandot `pwait -v <pid>` för att vänta på att `xterm` avslutar (ersätt `<pid>` med barnets processnummer). Gör samma sak igen, men avsluta `xterm` innan du kör `pwait`. Denna gång får du ett felmeddelande. I bägge fallen kommer föräldern aldrig att vänta på att barnet blir klart (avbryts innan `waitpid`). *Ge förslag på hur operativsystemet kan bära sig åt för att veta hur länge `waitpid` skall ge rätt resultat. Glöm inte att ta hänsyn hur det fungerade i föregående två frågor, speciellt fråga 2).*