

Styresystemer og multiprogrammering (OSM)

- G3

Dennis Bøgelund Olesen - 060593 - cwb759

Emil Lagoni - 051290 - frs575

Erik David Allin - 171292 - smt504

3. Marts 2013

Task 1

Task 1 a

I denne har vi implementeret stacken ud fra doubly linked listen som lavet i en tidligere opgave.

Dette er gjort af hensyn til at implementeringen ikke skal have de samme begrænsninger som den udleverede kode.

Ved at bruge dlist (doubly linked list) vil stacken være dynamisk. Derudover fungerer extract, ligesom pop, såfremt man altid indsætter og popper på hovedet af listen.

Som låse er der tilføjet en mutex på extract og pop, da disse ændrer i listen. Derudover er der lavet en condition variable der spinlocker empty() og top(). Denne sættes til at spinne ved start af push/pop og sættes til at stoppe efter kaldet. Dette gør at flere tråde kan kalde eksempelvis empty() samtidig, men derimod kan der ikke læses samtidig med at der pushes/poppes.

Task 1 b

I denne bruges strukturen void arg, til at sende argumentet til thread-create. Som det første hentes rowsne ind en ad gangen i stacken. Derefter sættes threadsne igang en ad gangen og popper fra stacken for at få argumenterne. I rowmult, som løser et row ad gangen, pushes til stack-result, hvor resultater så er gemt i, så vi senere kan stack-pop fra result, og indsætte i result matricen.

Hvis der er flere rækker end max antallet af threads, kaldes der join som venter på at tråden er færdig, og når de er færdige kan der laves nye tråde.

I denne opgave har vi ikke kunnet teste grundet forskellige fejl der har hindret os i at have et fungerende program.

Task 2

I denne har vi indskrevet funktionerne, semaphore-open, semaphore-v, semaphore-p og semaphore-destroy i `syscall.c`.

Til v og p, har vi udelukkende kaldet semaphore-p og semaphore-v fra kernel funktionerne. selve structen `usr-sem-t` er defineret som en `char const` og en `semaphore-t`.

I destroy er der gjort så den kun destroyer hvis semaphoren er ledig:

(*value* ≤ 0)

så den ikke ødelægger det for en tråd der venter.

I open bruger vi hjælpe funktionen `find-existing-block` til at tjekke om der i forvejen er semaphore med det givne navn. og `find-free-block` til at finde en ledig plads i `semlist`.

Denne kode er ikke blevet testet grundet problemer med compiling af test filer.