Styresystemer og Multiprogrammering

G-opgave #1

Jenny-Margrethe Vej
(rwj935@alumni.ku.dk)
Klaes Bo Rasmussen
(twb822@alumni.ku.dk)

Datalogisk Institut, Københavns Universitet
Blok 3 - 2013

A simple queue using a linked list

Vi har bygget koden op i samme rækkefølge som `queue.h`, og har ydermere lavet en `Makefile`, således man kan køre vores c-fil ved brug af kommandoen `make` i sin terminal.

(a)

Vores længde-funktion returnerer antallet af elementer i en kø. Den starter med længden 0, og er der 1 eller flere elementer, laves der en pointer til den nuværende node i køen, som vi kalder current, og lægger en til. Herefter går vi ned i while-løkken, som bliver ved at tjekke op på, om elementerne i køen er ens, lægger en til længden hver gang, de ikke er ens, og stopper så, når den rammer første element i køen igen.

`enqueue` tilføjer et element til enden af vores kø ved først at allokere pladsen til det vi gerne vil indsætte. Hvis det ikke er muligt at allokere den ønskede hukommelse, printer vi beskeden "Out of memory". I næste if-sætning, kigger vi på, om køen er tom, er den det, indsætter vi et element, som så vil pege på sig selv. Hvis der i forvejen er flere elementer, indsætter vi stadig det ønskede element, men opdaterer samtidig alle pegerne, så de står korrekt efter tilføjelsen af det nye element.

`dequeue` gør brug af samme tankegang som ovenfor, bare "med modsat fortegn" - vi fjerner altså et element her i stedet for at tilføje det, og sørger for også at frigøre pladsen fra det element, vi har fjernet.

Vi har testet alle 3 funktioner i `main()`, hvor vi først indsætter elementer i køen, for derefter at tjekke længden. Derefter fjerner vi elementer, og tjekker samtidig, at længde-funktionen også virker her.

(b)

Grunden til, at `enqueue` og `dequeue` skal have et argument af typen `QNode**` er, at når vi ændrer en værdi af `x`, ved at kalde en procedure, så skal vi bruge en pointer til `x`, som er adressen til `x` i hukommelsen. Kort fortalt er det fordi, at procedurer i C evaluerer efter strategien `call-by-value`.

(c)

Vores sum-funktion er bygget op på samme måde som `length`, med samme måde at løbe gennem køen på. Forskellen består selvfølgelig her i, at sum-funktionen summerer værdien af dataen fra køen. Funktionen er testet sammen med de andre funktioner, da vi alligevel skulle have nogle elementer i køen at teste ud fra.

`sum` tager 2 argumenter, og det ene satte vi til at være den indbyggede funktion `strlen`. Dog måtte vi lige lave en hjælpefunktion `mystrlen`, da datatypen eksplicit skal være en `char`, og ikke `Data`.

Buenos system calls for basic I/O

(a)

(b)

References