DIKUrevy 1987

Listigt nummer

skrevet af Står ikke

Status: Færdig

(n minutter)

Roller:

F (Står ikke)	Forelæser
H (Står ikke)	Head, et listehovede (blåt)
q (Står ikke)	Hjælpevariable (blå)
p (Står ikke)	Program (gult)
E1 (Står ikke)	Dynamisk element (rød)
E2 (Står ikke)	Dynamisk element (rød)
E3 (Står ikke)	Dynamisk element (rød)
E4 (Står ikke)	Dynamisk element (rød)
new (Står ikke)	(hvid)
gc (Står ikke)	garbage collector (sort)

Rekvisitter:

Overhead projektor ()
Et sæt paranteser, ca. 30 cm høje ()
Ring bind til noter og transparenter ()
Rengøringsvogn med kost og affaldssæk ()
4 hatte, f. ex. Bowler hatte ()

F : Goddag og velkommen. Sidste år lovede vi en forelæsning om kerner, men i år har vi byttet forelæsninger, så det nu er programmel. Jeg håber ikke I har forberedt jer forgæves, men I plejer jo slet ikke at forberede jer, så det gør vel ikke noget.

mnet i dag er listebehandling. Ved sidste forelæsning så vi på en stak, som var implementeret ved hjælp af en tabel, og vi så at der opstod problemer, hvis tabellen ikke var stor nok til at rumme stakken. I dag skal vi gennemgå en metode som lapper lidt på detet problem uden dog at løse det principielt. Og hermed er vi så kommet til hægterne.

Lad os starte med et simpelt eksempel: den tomme liste. Den tomme liste udmærker sig ved ingen elementer at indeholde, men for

at vide hvor vi har den, må vi have et listehovede til at pege på den. Må jeg få et listehovede ind!

programmet henter et listehovede, som lister ind

listehovedet peger ud mod publikum

F : Vi ser programmet hente et listehovede, og for at vise at listen er tom, skal hovedet sættes til NIL. I øjeblikket peger det på NILs Andersen. Bemærk at hovedet er helt blåt; det er fordi det er en statisk variabel.

programmet stikker hovedets højre hånd i dets lomme

F : Vi er nu klar til at indsætte et element. Det gøres naturligvis ved at sætte hovedet til at pege på elementet. Men først skal vi oprette det ny element. Elementerne tages fra en pulje efterhånden som behov opstår. Må jeg få en pulje ind!

ind kommer en pulje bestående af 4 dynamisk elementer

F : Man siger at elementerne oprettes dynamisk. Det er derfor at de er røde. Indtil de bliver oprettet befinder de sig i puljen og kaldes fri elementer. Vi ser her, hvordan de går frit omkring.

Oprettelse af et element sker ved at kalde standard proceduren new som tager et frit element fra puljen og sætter en pointer (dans jagthund) til at pege på det. I vores eksempel er det listehovedet, som skal pege på elementet.

Vi ser her hvordan programmet sætter parentesen om hovedet og kalder.

hovedet sætter parenteserne om hovedet og kalder:"new"

new kommer ind, tager et frit element E1, som begynder at græde. new tager hovedets arm op af lommen og sætter den til at pege på E1.

exit new

F : Vi ser at det gik helt fint at prøve at indsætte et nyt element.

programmet sætter parenteser om hovedet og kalder new. new kommer ind næsten som før, tager fat i E2, som har hånden i lommen. E2 begynder at græde. new sætter head til at pege på E2, hvorved E1 tabes på gulvet, bogstaveligt.

exit new

F : Hovsa det var ikke godt. Hvem gjorde det?

head og E2 peger på hinanden og siger i munden på hindanen: "det var ham"

F : Vi har nu opdaget den cykliske liste, og som ved alle væsentlige videnskabelige opdagelser skete det ved et uheld. Den cykliske liste er en meget vigtig struktur, som vi skal vende tilbage til ved en senere

lejlighed, men den kan nu ikke løse dette problem. Her bliver vi nødt til at rette fejlen i programmet

Vi indfører hjælpevariablen q.

E2's højre hånd stoppes i lommen

F : Vi ser nu hvorledes det går for sig. Jeg vil gerne pointere at hjælpevariablen q betragtes som statisk selvom den er lokal, og derfor også er blå.

programmet går ud og henter q. q spørger head hvem han peger på, og peger selv på den samme

programmet sætter parenteser om head og kalder: "new". Unter new, som tager E3 og sætter head til at pege på E3. Prorammet spørger head, hvem han peger på, og tage dennes, dvs E3's hånd. Programmet spørger så q, hvem han peger på og sætter E3 til at pege på denne (E2). Programmet går ud med q og kommer tilbage.

F : Ja, det var jo meget bedre. Vi tager det lige en gang til så alle er med.

det hele gentages, denne gange med E4 i stedet for E3

F : Sådan ja. Men det var jo en stak som var vores egentlige problem og foreløbig har vi skubbet det mere poppede foran os. Nu skal vi til at udtage elementerne igen. Læg mærke til vordan de sisdte her bliver de første. Det er en karakteristisk stakegenskab. Vi udtager topelementet – det som hovedet peger på – ved følgende algoritme.

algoritmen vises på overhead

på overhead: head:= head ↑ .next

programmet skal prikke til head og antyde at head skal pege på det som E4 peger på. Head skal spørge E4 hvad denne peger på, og pege på dette, som er E3. E4 tabes på gulvet, bogstaveligt.

F : Ja, her ser vi det igen. Et element er blevet tabt på gulvet. Hvis man vil bruge elemnetet igen, skal man huske at pege på det før det hægtes ud af listen; man kan sågar have en liste med elementer der ikke bruges og også puljen kan laves på denne måde. Men i dette tilfælde skal vi ikke umiddelbart bruge elementet til noget, så vi lader det ligge.

Vi forestiller os nu at programmet kører videre, og at det igen ønsker at føje et element til stakken. Vi ser hvorledes dette går for sig, nu da puljen af fri elementer er tom.

programmet henter q. q spørger hvad head peger på, og peger på det same. programmet sætter parenteser om head og kalder"new". Enter new, som undersøger puljen og ser at den er tom, og kalder garbage kollektoren. Enter garbage kollektor med rengøringsvogn

F : New har nu konstateret at puljen er tom, og har kaldt garbage kollektoren for at få eventuelle tabte elementer tilbage i puljen. Garbage kollektoren vil nu indlede sin første fase: mærkningsfasen, hvori de aktive elementer mærkes. Garbage kollektoren starter med at mærke de statiske variable og alle de dynamiske som de peger på, og igen de som disse peger på og så videre og så videre, indtil vi har nået afslutningen – af mærkningsfasen.

garbage kollektoren påbegynder mærkningen af head, q og kompagni

F : Det kan gå at benytte denne simple mærkningsalgoritme fordi vi ingen cykler har i strukturen, der som vi ser er en ren kæde. Garbage kollektoren har nu gennmført mærkningsfasen og går over til komprimeringsfasen.

garbagekollektoren tager en stor kost og tjatter til E4, der sendes over i puljen. Garbage kollektoren rykker E3 og derpå E2 hen mod head og justerer hænderne. Derefter får E1 samme behandling som E4, og også guitristen forsøges indsamlet

F : Nej, hov.

programmet tager sig til hovedet i panik. garbage kollektoren standser og protesterer

den peger på guitaristens hovede som er uden hat

F : Ja, godt nok er dette element ikke mærket, men garbagekollektion omfatter kun programmets egne variable. Garbagekollektoren har nu udført sin mission, og vi ser at new, som skulle oprette et element, fortsætter ufortrødent.

garbage kollektoren giver new et håndtryk og forlader scenen. new tager den nylige frie E4 og sætter head til at pege på det. exit new. programmet spørger head hvad denne peger på, og tager dettes, dvs E4's hånd. Programmet spørger q, hvad denne peger på, og sætter E4 til at pege på dette, E3. programmet følger q ud.

F : Her vil jeg slutte gennemgangen af de hægtede stakke. Klokken er også blevet mange. Næste gang skal vi se på hægtede multilister i netværksdatabaser. Tak for i dag.

programmet kommer igen ind og få alle ud af scenen