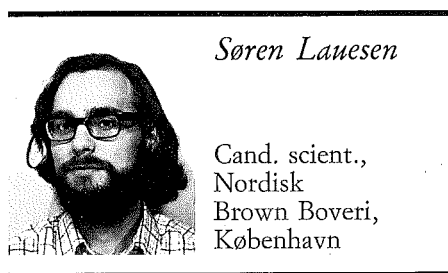


Kombineret processtyring, time-sharing og batch på Kemisk Institut, Århus Universitet.

Indledning

Den 5. oktober 1972 afsluttede A/S Regnecentralen afleveringsprøven af en RC4000 datamat på Kemisk Institut, Århus Universitet (i det følgende kaldet KI). Som en del af prøven skulle anlægget køre 3 uger med højst 4 systemsammenbrud i alt på grund af hardware eller software. Der skulle i perioden køres processtyring, time-sharing, batch på én gang og programmerne skulle samtidig kunne udvikles og ændres af KI.

Denne artikel beskriver opgavetyperne, systemets hardware og software, samt pålidelighedsovervejelserne for software.



Søren Lauesen

Cand. scient.,
Nordisk
Brown Boveri,
København

Systemets opgaver

Systemet tilpasser sig dynamisk til varierende driftsformer som krævet af de tilmeldte job. I et givet øjeblik kan systemet på KI typisk udføre følgende job parallelt (se fig. 1):

1. To procesjob, der samler måleresultater op hvert sekund og kører flere døgn.
2. Et procesjob, der samler måleresultater op hver 160 mikrosek. og kører omkring en time.
3. Tre time-sharing job tilknyttet hver sin terminal.
4. Et antal job tilmeldt fra skrivemaskine-terminaler som remote job.
5. Et antal batch job.
6. Et transmissionsjob, der på opfordring fra de øvrige job transmitterer filer til CDC 6400, modtager resultater derfra og får dem trykt.

Alle procesjob og andre brugerjob er skrevet udelukkende i Algol eller Fortran, men maskinkode-programmer ville ikke ændre systemets sikkerhed.

Hardware

Anlægget består af en RC4000 centralenhed med 32 k lager (96 k tegn), et pladelager på 6 mill. tegn med bevægelige hoveder (accestid ca. 100 ms), én båndstation, strimmellæser, linieskriver, plotter, telekommunikationsenheder tilsluttet af KI selv. Systemets ydeevne kunne øges væsentligt ved anskaffelse af en tromle med faste hoveder (380 k tegn) og gerne også et ekstra lagermodul (48 k tegn).

Software

Systemet er baseret på RC4000's multiprogrammerings-monitor (ref. 1), der tillader parallel udførsel af en række programmer. Når et program udføres, råder det over et sammenhængende areal i ferritlageret og kaldes en *proces* (job eller task). Processerne er totalt beskyttet mod hinanden af hardware og software, således at en programmeringsfejl i den ene ikke kan forstyrre den anden. Alligevel har processerne mulighed for at kommunikere med hinanden – hvis begge parter er indstillet på det.

Nogle processer kan opfattes som brugerjob, andre som operativsystemer, og andre igen som komplicerede drivere af ydre enheder. En proces kan bruge en del af sit lagerareal til at oprette en

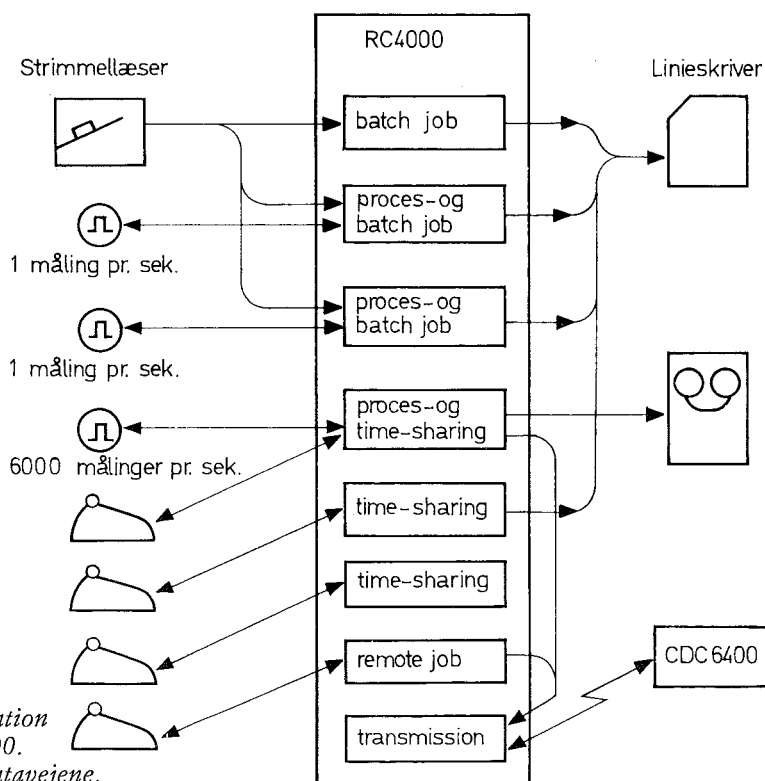
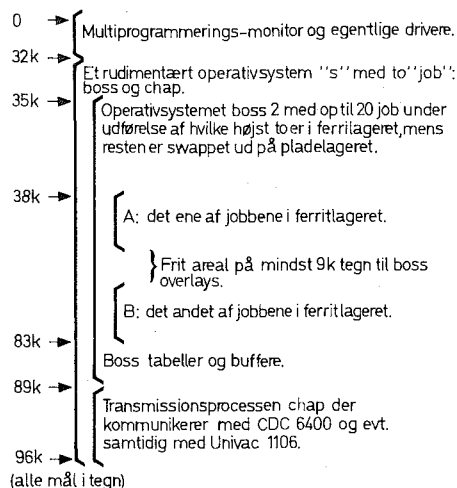


Fig. 1.
Typisk jobsituation
på KI's RC4000.
Pilene viser datavejene.

eller flere parallelt kørende under-processer. På KI kan systemet af processer anskueliggøres med følgende ferrit-lagerbillede:



Som det ses er det samlede lagerareal, der er til rådighed for de to brugerjob, A og B 36 k tegn, hvilket er tilstrækkeligt til to job af standardstørrelsen 18 k. I et sådant areal kan man oversætte vilkårligt store Algol- og Fortran-programmer, idet pladelageret benyttes til mellemresultater. Oversættelsestiden er på KI ca. 200 inputtegn pr. sek., hvilket ville øges til 1000 tegn pr. sek., hvis en fasthoved tromle blev installeret. Man kan ligeledes køre vilkårligt store oversatte programmer, når blot de variable (arrays og dimensions) kan være i lageret, idet programmerne automatisk kører med software paging (virtuelt lager).

Hvis et af jobbene (f. eks. A) har brug for mere lagerplads, vil det andet (B) blive swappet ud, så A kan disponere over alle 36 k tegn.

Administration af jobtyperne

Vi vil nu forklare, hvordan de forskellige jobtyper i afsnit 2 passes ind i den generelle ramme ovenfor.

Type 1: Konverserende procesjob

Disse job styrer forholdsvis langsomt procesudstyr. Jobbet ligger på plads A i lageret og starter procesudstyret, hvorefter måleresultaterne opsamles af monitoren driver (ét interrupt pr. måleresultat) og lagres i en buffer i Boss (fig. 2). Opsamlingen af en buffer strækker sig typisk over et minut, og i denne periode har Boss swappet jobbet ud på pladelageret, således at andre job kan udføres på plads A. Når det ønskede antal måleresultater er indlæst (evt. bestemt af bufferstørrelsen) swappes jobbet ind igen, behandler resultaterne og bestemmer, hvad der videre skal ske, f. eks. indlæsning af flere resultater eller programstyret justering af apparatet. Hele kørselen tager indtil flere døgn.

Programmet i jobbet er i praksis skrevet i Algol uden brug af maskinkode, men det kunne selvfølgelig skrives i assemblersprog uden at sikkerheden over for systemets øvrige job ville lide under det. Driveren i monitoren er skrevet i maskinkode (af KI selv) og oversat samtidig med monitoren. Den er meget simpel og behøver ikke at ændres, selv om jobbet skifter strategi i behandlingen af procesudstyret. Den del af kommunikationen, der ligger i Boss, er en standard-facilitet, der kan bruges ved mange andre ydre enheder i uændret form.

Vi vil fremhæve det fundamentale princip i denne processtyring: Styringen opdeles i en højfrekvent del (opsamlingen af måleresultater) og en lavfrekvent del (den strategiske behandling af en buffer af resultater). Den højfrekvente del er her placeret i monitoren, der kan klare kravene til responsetid. Den lavfrekvente del har langt mindre krav til responsetid, men større krav til lagerplads. Den kan derfor placeres i et job under en swapping strategi hos operativsystemet.

Type 2: Lagerlåste job

Disse job styrer meget hurtigt procesudstyr. På KI er der tale om en film-scanner, der afleverer et måleresultat pr. 160 mikrosek. For hvert 2000. resultat er der en pause på 50 msek. I alt tager hele kørselen omkring en time. Hvis ferritlagerkravet skal være nogenlunde rimeligt, kan en sådan enhed ikke styres af et job, der swappes. Indlæsningen af et måleresultat kan heller ikke klares på interruptbasis med rimelighed. Vi har her valgt den løsning at lade en blok på 2000 måleresultater indlæse over high-speed datakanalen (en multiplexer kanal) der mageligt kan følge med, selv om bånd og pladelager kører simultant. Når blokken er indlæst, behandles den af jobbet, der ligger permanent i lageret på plads B under hele kørselen. Da 50 msek ikke er tilstrækkeligt til at behandle blokken, kører jobbet med dobbelt buffer, således at der er 370 msek til rådighed. Vi har her igen en opdeling i højfrekvent og lavfrekvent behandling, denne gang med den højfrekvente del i en hardware controller og den lavfrekvente i et job låst fast i lageret.

Denne kørselsform administreres ved at jobbet ved målingernes start beder

Boss om at blive låst i lageret et vist stykke tid. Hvis de konverserende job på plads A kan være i lageret samtidig med låsejobbet, kan Boss straks sikre anmodningen. I modsat fald er der endnu mulighed for at forsinke låsejobbet. Hvis dette skal køre acceptabelt i praksis, må der være en aftale mellem brugerne om at procesjob altid har standard størrelse (så længe lageret er så lille). En sådan aftale kunne om nødvendigt kontrolleres af Boss via brugerkataloget, men det har ikke været nødvendigt, da standardstørrelsen tillader alle nødvendige operationer som oversættelse og udførelse.

Type 3: Time-sharing job

Disse job konverserer med brugeren via en skrivemaskine-terminal under kørselen. De ligger på plads A og swappes ud, når brugeren taster input og medens terminalen skriver output. De ligner i mange henseender konverserende procesjob, hvor terminalen er »procesudstyret«. Der er dog en lille strategisk forskel, idet output til terminalen spooler over pladelageret for at mindske antallet af swap, når der produceres meget output.

Type 4: Remote job

Et time-sharing job beslaglægger ganske vist ikke ferritlageret mens konversationen står på, men det beslaglægger andre ressourcer, f. eks. plads på pladelager, tabeller og buffere i monitoren. Desuden er swapping teknikken ret tung og kræver meget arbejde af monitor og Boss.

En nøjere analyse viser at behovet for direkte konversation er størst i forbindelse med editering af tekster, mens oversættelse og udførelse normalt kan ske helt off-line. Boss tillader derfor at en terminalbruger editerer og lister sine filer via en hurtig editor i Boss selv, uden at han har et kørende job. Når editeringen er fuldført, tilmeldes jobbet til kørsel og output spooler via pladelageret til terminalen. Typiske mindre job kommer på den måde til at beslaglægge ressourcer i et halvt minut, selv om editeringen tog et kvarter.

Remote job kan, afhængigt af deres køretid, udføres på plads A eller B i ferritlageret. De bliver swappet ud, hvis et konverserende job eller et højere prioriteret job har brug for lageret.

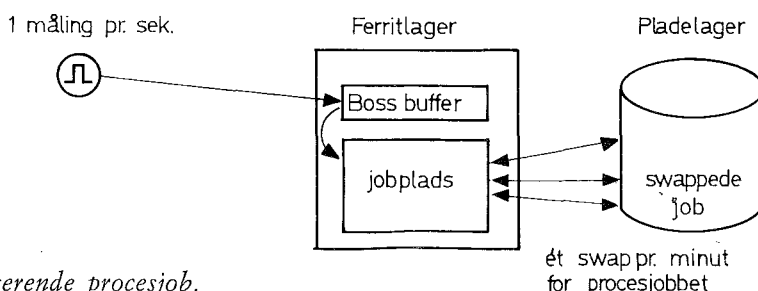


Fig. 2. Konverserende procesjob.

Type 5: Batch job

Job kan indlæses og tilmeldes via strimmellæser eller kortlæser (KI har dog endnu ingen kortlæser) og de behandles da analogt med remote job, idet output spooler via pladelageret til linieskriveren. Endvidere kan ethvert job tilmelde batch job til kørsel, således at man f. eks. fra en terminal kan sende lange job til batch kørsel.

Type 6: Transmission

Ethvert job kan give transmissionsprocessen »chap« besked om at sende en fil (på pladelager) til CDC 6400 på RECAU (eller, hvis KI ønskede det, til UNIVAC 1106 på RECKU). Chap styrer transmissionen af en kø af filer og modtager output fra det andet (de andre) anlæg. Output kan kun trykkes via Boss, der beslaglægger printeren for sig og sine job, men det klares ved at chap tilmelder et passende batch job til kørsel under Boss.

I den foregående forklaring har vi ladet som om et job kunne klassificeres absolut i én af båsene, men det er ikke nødvendigt. F. eks. kan et job godt være et remote job og et konverserende procesjob samtidig, eller et time-sharing job og et lagerlåst job samtidig (så udføres det på plads B). Den eneste kombination, der ikke tillades, er at et batch job forsøger at bruge en terminal.

Øvrige operativsystem-faciliteter

Trods det ringe pladskrav indeholder systemet de faciliteter, man normalt forventer af et operativsystem.

Der er således et generelt filsystem, der skelner mellem permanente filer, der skal kunne overleve selv alvorlige hardware-fejl og temporære filer, der kun skal overleve et job. Alle filer på baggrundlager (tromle eller pladelager) kan adresseres random og udvides/indskrænkes vilkårligt uden ekstra omkostninger. Der findes et hierarkisk katalog over filerne, hvor bl. a. hver bruger har sit private filkatalog.

Der er et avanceret jobstyresprog med betingelser og mulighed for brugerdefinerede kommandoer og hjælpeprogrammer. Ydre enheder kan behandles på standardmåde, eller med brugerindgreb i fejlsituationer, eller fuldstændig brugerstyret, eller brugerstyret undtagen i fejlsituationer.

Der er dynamisk prioritering af jobbene og fuldstændig operativsystem kontrol med alle ressourcer – også baggrundslagerplads og filkatalogplads. Strategien i ressourcefordeling og prioritering er beskrevet i ref. 3.

Kun en brøkdel af faciliteterne vedrører operativsystemet Boss; det meste ligger i multiprogrammerings-monito-

ren eller i de programmer, der loades i jobbet's ferritlagerareal (det gælder f. eks. hele jobstyresproget).

Driftssikkerhed, udviklingsforløb

Det meste af systemet består af programmer, der har været i drift længe. F. eks. har monitor, oversættere, fortolker af jobstyresproget været i drift siden 1969, omend i en lidt afvigende version. Indtil 1972 har de fleste installationer kørt under det rudimentære operativsystem »s«, på en måde, der nærmest minder om IBM's DOS.

Erfaringen fra denne periode viser, at systemets sikkerhed er meget høj, specielt er monitor og »s« i stand til at overleve enhver programmeringsfejl i jobbene eller oversætterne. Typisk bryder systemet ned en gang hver 3. måned (som regel på grund af en alvorlig hardwarefejl), hvilket skal sammenlignes med de regionale centres opstarter flere gange daglig.

Ved konstruktionen af Boss 2 (startet sidst i 1970) var det derfor et mål at opnå en tilsvarende driftssikkerhed 1/2 til 1 år efter frigivelsen af systemet. Der er derfor ikke gjort noget forsøg på at retablere jobkøen, spoolarealer etc. efter et sammenbrud. Kun de permanente filer er sikret ved en omhyggeligt planlagt, men yderst enkel katalogstrategi.

For at kunne opfylde målet, må vi altså kunne lokalisere fejl, der optræder en enkelt gang under praktisk drift. Det er sikret ved at Boss under driften udskriver detaljeret testoutput om alle sine aktiviteter på komprimeret binær form – enten på et magnetbånd eller på en baggrundslagerfil, der udnyttes cyklisk. To timers normal drift kan let fylde en stor båndspole med testoutput. Får vi tilsendt et bånd med testoutput har vi altid kunnet lokalisere fejlen. Er der kun en baggrundslagerfil til rådighed, kan fejlen som regel lokaliseres, især hvis den er sket i den sidste tidsperiode.

Systemet blev sat eksperimentelt i drift på tre installationer i maj 72. I august 72 kunne det frigives officielt. I oktober 72 afsluttedes afleveringsprøven på KI, hvor systemet kørte 3 uger i døgn-drift med 4 sammenbrud på grund af software (Boss 2). Der var ingen restriktioner på jobtyperne og den belastning, anlægget kunne udsættes for. Faktisk blev der foretaget processtyring, programindkørsel osv. med en spidsbelastning på ca. 100 job fra kl. 11 til 16 og med en bogført jobkøretid på op til 22 timer i døgnet.

Systemets fremtid

Den oprindelige specifikation af Boss 2 anførte en række udvidelser, der ikke skulle implementeres i første omgang.

I forbindelse med leverancen til KI, blev nogle af disse udvidelser implementeret samt nogle, der ikke tidligere havde været overvejet (processtyring, transmission til CDC 6400). Disse udvidelser skete meget let i foråret 72 og det er vores overbevisning, at systemets grundstruktur er tilstrækkelig simpel og generel til at bære langt flere udvidelser, f. eks. med egentlige remote batch terminaler (hurtige) og data entry systemer til administrativ databehandling. Trods den manglende Cobol-oversætter anvendes systemet med succes i Regnecentralens egne servicecentre til administrativ databehandling, idet Algol og Fortran er udvidet med faciliteter, der gør dem velegnede til formålet.

Fagligt set må systemets software anses for konkurrencedygtigt i mange år fremover, især hvis man ofrede udvikling af Cobol eller PL1. Hardware kunne ved en beskeden indsats tilpasses til moderne minicomputer-teknik og bevares konkurrencedygtigt over den samme årrække. Desværre kræver markedsføring jo andet end et godt produkt, og Regnecentralens nylige strukturændring betyder, at man ikke fremover vil satse på dette felt.

Software-gruppen

Boss 2 er designet og implementeret af Klavs Landberg, Per Mondrup og forfatteren, senere med deltagelse af Bjørn Ørding Thomsen, Bo Tveden Jørgensen og Bo Jacoby. Forfatteren ledede projektet indtil juni 72, Klavs Landberg derefter. Tilpasning af det øvrige software er udført af Peter Lindblad Andersen og Hans Rischel.

Transmissionsprocessen er designet og implementeret af Ole Caprani, Lise Lauesen og Flemming Sejergård Olsen, alle Datalogisk Institut, Københavns Universitet.

Vi har været meget glade for det kvalificerede samarbejde med medarbejderne ved Kemisk Institut i den lange fejlfindingsperiode fra maj til september 72. Specielt vil vi takke Finn Rasmussen for en usædvanlig evne og vilje til at isolere fejl og rapportere de relevante oplysninger. ■

Ref. 1: Per Brinch Hansen: The nucleus of a multiprogramming system. Comm ACM 13, 4 (april 1970), p. 238–241, 250.

Ref. 2: Søren Lauesen: Boss 2, User's Manual, Operator's Manual, Installation and Maintenance. RCSL No.: 31-D211, 31-D123, 31-D191, Regnecentralen 1972, København.

Ref. 3: Søren Lauesen: Job scheduling guaranteeing reasonable turnaround times. Acta Informatica 2, 1–11 (1973).