

3 februari 1965

Communicatiebuffering voor de EL-X8 - T.H.E.

Algemeen

De mogelijkheid communicatie-informatie te bufferen is een faciliteit van de installatie als geheel: de bufferruimte wordt dan ook niet bij speciale programma's ondergebracht, maar "universaliter", zoals de tabel van de SV's van de bibliotheek.

Dit doet ons leed, want het impliceert een permanente kerngeheugenbezetting voor de SV-lijst van de buffersegmenten. Toch moet het maar; wij denken aan, zeg 256 SV's (d.w.z. maximaal $\frac{1}{4}$ trommel vol gebufferde transport informatie).

Naast de 256 SV's zullen wij extra ruimte ter beschikking stellen om hierin kettingen van segmenten te rijgen.

Analyse van de Segment Controller heeft aangetoond, dat het niet ondoenlijk is om SV's te verhuizen; dit maakt het mogelijk, dat een producent een segment als privé segment vult, en als het vol is de informatie naar de bufferruimte overhevelt, zonder de informatie "vleselijk" te transporteren. Er hoeft slechts een SV verhuisd te worden.

Invoer via bandlezers

De invoer via bandlezers behandelen we apart, omdat we deze tamelijk afwijkend bedienen.

De Pm's nemen niet rechtstreeks contact met de bandlezers op, maar doen dit via een informatiestroom, waarvan de opbouw ver op de consumptie vóór mag zijn. Men kan denken dat de bandlezer zo snel is, dat dit nauwelijks de moeite waard is; wij moeten dit echter wel doen, omdat de buffering de enige manier is om de bandlezer zo gauw mogelijk vrij te maken.

Elke Pm krijgt slechts de beschikking over een enkele papierband-informatiestroom; het feit dat er (wellicht) meer bandlezers aan de installatie zitten komt niet de individuele programma's, maar het samenspel ten goede.

Om deze impliciete invoerestroom hoeft een programma niet te vragen: hij was er kennelijk al, toen het programma ingevoerd ging worden.

Wij zagen twee wegen: of de bandlezers in de bankiersalgorithme betrekken en van de programmeur verlangen, dat hij de interesse "afmeldt" wanneer hij zijn laatste gegeven heeft ingelezen of het er maar op wagen, d.w.z. niet de programmeur laten verklaren dat hij niet meer lezen wil, en, omgekeerd, de bandlezers buiten het bankiersalgorithme houden. Wij hebben - tegen alle regels in! - tot het laatste besloten. Om het ruimtebeslag in de buffer te mitigeren hebben we wel besloten, om in de informatiestroom van de lezers het bandbeeld gepakt te bergen. (Een pakking van 2000 karakters per segment geeft bij volle band van 300 meter à 400 kar. per meter 60 segmenten. Dat is natuurlijk niet mals, maar het zij zo.)

Wij hebben dit onder anderen gekozen, omdat wij niet veel vertrouwen hebben in de zo moeilijk afdwingbare plicht tot afmelden. (Wij doen ons best onze programmeurs niet te hoog aan te slaan!)

Reservering van een bandlezer voor de opbouw van een bepaalde informatiestroom valt onherroepelijk af bij de detectie van "einde band".

Deze informatiestromen zijn objecten, die elk behoren bij een bepaalde PM. We kunnen ons het "handvat" onderin de stapel van de PM voorstellen. (Geheugenprotectie schept wel problemen, want de producent moet bij het handvat kunnen, maar de producent is tenslotte een stamgast, die moet weten wat hij doet.)

Over de gedetailleerde layout van het handvat doen we nu geen voorstel. Wij merken op, dat er onderscheid gemaakt moet worden tussen "informatiestroom nuchter" - d.w.z. niet eens in opbouw - en "informatiestroom leeg", d.w.z. het komt wel, maar er zit geen enkel segment in de buffer-ruimte. Het handvat bevat de seinpaal, via welke de producent toevoeging van één nieuw segment aan de consument doormeldt. Het handvat bevat tevens een identificatie van de producent - om vrij te kunnen maken in geval van desaster, b.v.

Plotter, printer & ponser

Wij nemen aan, dat een plotter, die aan een bepaald plaatje begonnen is, dit plaatje moet kunnen afmaken, voordat hij aan een ander plaatje begint. Voorts stellen wij, dat een plotterprent potentieel zo uitgebreid kan zijn, dat wachten met output, het programma "einde plaatje" heeft gemeldt, prohibitief zou kunnen zijn. Hieruit volgt, dat met 1 fysieke plotter, geen PM via twee "abstracte" kanalen, twee - willekeurig uitgebreide - plaatjes kan gaan tekenen. Wij voorkomen dit door geen PM het uitdrukingsvermogen te geven om aan meer dan een plaatje bezig te zijn. Bij voorkeur zal de plotter (zie onder) slechts aan plaatjes beginnen, waarvan de specificatie volledig voltooid is.

Ook voor de bandponzers geldt, dat we bij voorkeur slechts een ponser aan het ponsen van een bandje zullen laten beginnen, als het bandbeeld geheel is opgebouwd. Als dit nl. zo is, kan de machine controleren, of er nog genoeg papier op de rol is. Moet een ponser aan een onaf bandbeeld beginnen, dan zal hij vragen om een nieuwe band in de ponser.

Tenslotte: de highspeed printer verwerkt van de programma's de informatie "formuliersgewijs". Als echter twee programma's met dezelfde snelheid formulieren aanbieden is het voor de receptie hoogst onaangenaam, als de formulieren om de ander komen. Ook hier hebben we dus zoiets, dat de formulier printer - zo mogelijk - alle teksten produceren.

Ondanks de verschillen zullen wij proberen, deze drie apparaten zo gelijk mogelijk te bespreken.

Het volgende beeld werd in eerste instantie voor de plotter(s) ontwikkeld. Wij zullen straks bekijken, in hoeverre het ook voor de andere gereedschap van toepassing is. Wij praten dus eerst even over de plotter(s).

kwantitatief spelen twee aantallen een grote rol: het aantal fysieke apparaten en het aantal "documenten in opbouw". (Een "document" is voor een plotter een plaatje, voor een bandponser een noodzakelijkerwijze onverknippte band.)

Wij stellen ons voor, dat een fysisch apparaat niet aan het volgende document begint, voordat het vorige is voltooid. (Bij de plotter is dat niet strikt noodzakelijk!) Wij gaan er verder van uit, dat een document zo omvangrijk kan zijn, dat het onredelijk is om a priori te eisen, dat het in zijn totaliteit gebufferd moet kunnen worden, voordat aan de feitelijke uitvoer begonnen wordt. (Dit is voor de ponsers weer wat restrictief, omdat de lengte van een fysieke band wel aan een bovengrens gebonden is.)

Voor een bepaald soort fysisch apparaat voeren wij in "de uitvoerders" - in evenveelvoud als er fysieke apparaten zijn - en "de prae-uitvoerders". Elke prae-uitvoerder is de mogelijkheid om een eventueel "onbegrensd" document te gaan opbouwen. Prae-uitvoerders zijn geprogrammeerde instituten, hun aantal kan in overigens identieke installaties verschillen.

De programmeur kent het verschil tussen de prae-uitvoerder en de uitvoerder niet. Als hij als zijn behoefte opgeeft b.v. "1 plotter", dan wordt deze behoefte ten bate van de bankier vertaald in "1 praeplotter + 1 plotter". Als hij aangeeft, dat hij een document wil gaan opbouwen, dan wordt dit vertaald in de poging om een prae-uitvoerder te lenen. Nu krijgt de PM een prae-uitvoerder toegewezen: hij kan nu het document opbouwen - mogelijk hierin geremd door een P-operatie op vrijkomende bufferruimte en tenslotte wordt van hem verwacht, dat hij het document "voltooid" meldt. Op dit moment komt de prae-uitvoerder vrij en kan een andere - of dezelfde - PM aan documentopbouw beginnen.

Als er n prae-uitvoerders en m echte uitvoerders zijn, spelen achter de schermen $n + m$ overigens equivalente informatiestromen een rol: hiervan zijn op elk moment een ten hoogste n -tal "de prae-uitvoerders". (Deze som komt voort uit het volgende: elke fysieke uitvoerder kan bezig zijn met de consumptie van een inmiddels af document; daarnaast zijn n documenten in opbouw.)

Als een PM een document voltooid meldt, dan moeten wij twee getallen onderscheiden: of een feitelijke uitvoerder was reeds aan de consumptie van deze stroom begonnen of niet. Zo nee, dan wordt het voltooide, onbegonnen beeld aan de z.g. "documentenketting" gehangen. Naast genoemde informatiestromen die hoogstens één document in opbouw, in afbraak of beide bevatten, is er nl. nog een ketting van ~~ook~~ voltooide documenten. In deze ketting worden klare documenten opgenomen in volgorde van voltooiing; in deze zelfde volgorde zal later aan hun feitelijke uitvoer begonnen worden. (N.b.: Bij meer uitvoerders betekent dit niet, dat ze de een na de ander in volgorde uitgevoerd hoeven te worden!) Verder wordt er een prae-uitvoerder vrijgegeven.

Als bij voltooiing van het documentbeeld wel reeds een fysieke uitvoerder aan de feitelijke consumptie begonnen was, wordt de documentenketting met rust gelaten, maar wordt aan de bankier beëindiging van lening gemeld van niet alleen een prae-uitvoerder, maar ook van een uitvoerder. (Zie onder.)

Een vrij ledige uitvoerder handelt als volgt:

Als de documentenketting niet leeg is, zoekt hij uit de $m + n$ informatiestromen een ongebruikte (die is er!) en initieert deze stroom met het kopdocument uit de ketting. Vervolgens gaat hij uit deze stroom consumeren.

Als de documentenketting leeg is, kan hij, wanneer er wel documenten in opbouw zijn, proberen, of hij al vast aan de uitvoer van een onaf document beginnen kan. Het beginnen aan een onaf document betekent echter dat het opbouwende programma vanaf dat moment geacht wordt behalve een prae-uitvoerder ook een uitvoerder geleend te hebben. De uitvoerder mag zich dus slechts onder controle van de bankier aan een onaf document beginnen.

Als een uitvoerder zichzelf uit de documentenketting initieert, is dit een aflopende taak: de uitvoerder kan zich op daggeldbasis ter beschikking stellen en omarmingsrestricties worden hierdoor aanmerkelijk milder.

het idee van de prae-uitvoerders is dus om zo mogelijk op daggeldbasis van de fysieke uitvoerders gebruik te maken; dit is des te interessanter naarmate de beschikbare bufferruimte groot is ten opzichte van voorkomende document ontvangen.

Ben en ander werpt wel strategische problemen op, speciaal: hoeveel prae-uitvoerders kiezen we en wanneer beslissen we, dat een uitvoerder begint aan de uitvoer van een onaf document. (De mogelijkheid zulks te doen wordt door de bankier afgeperkt: als het mogelijk is, hoeft de uitvoerder het echter nog niet te doen.)

Ben enkel voorbeeld moge de strategische problemen nader illustreren. Stel, dat we in een installatie van een bepaald type 1 uitvoerder hebben, maar meer prae-uitvoerders en stel, dat de installatie geladen is met een programma, dat een niet ontegenwoordig omvangrijk document heel traag vult (een z.g. "druppelaar"). Zolang de omvang van de gebufferde informatiestroom nog geen onbehoorlijke proporties heeft aangenomen, is het zaak, de uitvoerder nog niet aan de uitvoer te laten beginnen: dit kan de fysieke uitvoerder nl. nodeloos lang bezig houden. En inderdaad: terwijl de druppelaar moeizaam zijn document opbouwt, komt een extra programma, een "vluggertje", die eventjes een snel produceerbaar document wil uitvoeren; van de kant van de machine is dit een ideale "mix". Het tweede programma gaat document produceren: als nu dit tweede document - b.v. omvangrijk - niet eerst in zijn geheel gebufferd kan worden, moet het fysisch apparaat wegens gebrek aan bufferruimte op een gegeven ogenblik aan een van beide documenten beginnen. Hoe besturen we, dat hij dan aan het document van het vluggertje begint?

Er zijn twee oplossingen: als een uitvoerder aan een onaf fysisch document moet beginnen - wegens uitputting van de bufferruimte - laat hem dan in geval van keuze de langste informatiestroom kiezen: dit helpt het meeste. In boven beschreven situatie moet je dus hopen dat op het moment dat tot uitvoer besloten wordt, de producten van het vluggertje de productie van de druppelaar inmiddels heeft ingehaald. (Je moet dus met het vluggertje wel "op tijd" komen: dat moet toch, want als de druppelaar definitief aan de uitvoerder gekoppeld is, dan heb je het gehad en komt het vluggertje toch al niet meer aan bod. Deze inhaaloverweging stelt de grens van "op tijd zijn" alleen wat scherper.)

er is een andere techniek, nl. die der vrijheidsklassen. Hierin kent men aan elk programma een vrijheidsnummer toe: de bankier hanteert nu als nevenvoorwaarde, dat een programma met hoger vrijheidsnummer voor zijn voortzetting niet op de voortzetting van - d.w.z. vrijgave door - een programma uit een lagere vrijheidsklasse afhankelijk mag zijn. Door

de druppelaar dan een lager vrijheidsnummer te geven dan het vluggertje forceert men, dat

- a) het vluggertje alleen geaccepteerd wordt als de uitvoerder nog niet aan het document van de druppelaar begonnen is,
- b) daarna de feitelijke uitvoer van het document van het vluggertje de voorrang krijgt.

Of de invoering van het vrijheidsgetal geen nadere complicaties met zich meebrengt - afgezien ervan, dat men het moet definiëren - moet een nadere analyse nu uitmaken.

Zonder invoering van het vrijheidsgetal kan men altijd (logisch) straf-feloos een vrije uitvoerder aan minstens een van de onaffe documenten laten beginnen. Immers: dit geldt als ~~er~~ een lening van de uitvoerder. Voordat deze koppeling tot stand gekomen is, was de toestand veilig, d.w.z. de bankier zag al zijn programma's op den duur eindigen, d.w.z. kon dit in een zekere volgorde verifiëren: het eerste prae-pensende programma, waarvan hij de doorgang garanderen kan, mag dus ook de feitelijke uitvoerder op dit moment willen lenen. Houdt ook na introductie van het vrijheidsgetal deze redenering stand?

In dat geval onderzoekt de bankier eerst of de programma's uit de hoogste vrijheidsklasse door kunnen gaan. Hieronder kunnen er zijn, die wel prae-uitvoerder en uitvoerder geclaimd, maar nog niet geleend hebben. Vervolgens worden de programma's uit de daaronder liggende klasse onderzocht: hier kan het eerste prae-uitvoerende programma ontmoet worden, dat evenwel de uitvoerder zelf niet lenen mag, omdat dan een programma uit hogere vrijheidsklasse in de knoei zou kunnen komen.

Moraal is: zonder vrijheidsklasse kan altijd de uitvoerder aan minstens 1 niet leeg onaf document beginnen. Met vrijheidsklasse - en vrije prae-uitvoerders - is het mogelijk de uitvoerder voor een nog lege prae-uitvoerder te reserveren!

Het voorgaande dekt vrij aardig de bandponser en de plotter: de high speed printer is een ander chapter.

Om te beginnen: het lijkt irreal om voor de printer het aantal prae-uitvoerders lager te kiezen dan het aantal PM's ("een programma, dat niet print is nauwelijks een programma").

Als wij ons op het standpunt stellen, dat de printer snel is, dat de programmeur snel zijn gegevens naar buiten wil hebben, dan zou men zeggen: beschouw een formulier van de printer als "document", zodra dit klaar is, neem het dan in de ketting op. Anderzijds is dit onacceptabel: het scheur- en sorteerwerk, dat hierdoor voor de expeditie zou ontstaan is gewoon te erg. En dan ga je dus zeggen: beschouw aan elke prae-uitvoerder "N" formulieren als document. Maar hoe groot kies je "N"?

Het is duidelijk, dat beëindiging van het programma de prae-uitvoerder vrijgeeft en het startje formulieren als document naar de ketting kan overhevelen. Dan is er ten duidelijkste werk voor de printer. Als de documentenketting leeg is, begint de printer (beginnen de printers) pas actief te worden als de totale omvang van de gebufferde informatie daartoe aanleiding geeft. Het ligt voor de hand, om dan uit een informatiestroom, die begint met het oudste document of de informatiestroom, die het grootste aantal formulieren bevat? Mijn neiging is om de tijdsfactor te verwaarlozen - er komen geen programmeurs in de machinekamer - en de langste informatiestroom te pakken.