X8 nr. 21.

Notitie over de aansluiting en de programmering van de teleprinter.

## 1. Hardware.

### 1.1. Hardware voor de teleprinter (dwz. outputorgaan).

De hardware bestaat uit:
een 2-waardige startseinpaal, genaamd "Acflop teleprinter",
een 2-waardige ingreepseinpaal, genaamd "Inflop teleprinter",
een vaste geheugenplaats, die fungeert als buffer met capaciteit van 1 karakter;
het vullen van deze vaste geheugenplaats zal ik in de tekst representeren door

"teleprinter:=....." .

Voor de bediening van het printwerk bestaat dus slechts het minimum: we kunnen geen startopdrachten accumuleren en per startopdracht kunnen we slechts 1 karakter aanbieden. De opmerking is, dat een teleprinter een zo langzaam medium is, dat alles, wat er meer ingebouwd wordt, moeilijk te verdedigen is.

Het aanbieden van een karakter aan de teleprinter geschiedt door de sequens:

"P(Inflop teleprinter);
teleprinter:= karakter;
V(Acflop teleprinter)" .

De P-operatie is wel een aanroep van de Coordinator, de V-operatie kan zonder meer in het programma staat (Nl. "Acflop teleprinter:= 1").

### 1.2. Hardware voor het toetsenbord (dwz. inputorgaan).

Hiervoor zijn twee voorstellen ter sprake geweest; het doel van deze notitie is onder andere om er achter te komen of het ene voorstel preferabel is boven het andere.

#### 1.2.1. Toetsenbord met expliciete "Acflop".

De hardware bestaat uit:
een 2-waardige startseinpaal, genaamd "Acflop toetsenbord",
een 2-waardige ingreepseinpaal, genaamd "Inflop toetsenbord",
een vaste geheugenplaats, die fungeert als buffer met capaciteit van 1 karakter;
het uitlezen van deze geheugenplaats zal ik in het volgende representeren door:

".....:= toetsenbord".

Hier wordt het toetsenbord dus beschouwd als een invoermedium zonder de mogelijkheid van accumulatie vanstartopdrachten en per startopdracht slechts 1 karakter.

Drukt men op de toets van het toetsenbord, dan stuurt de toetsapparatuur een startpuls, vijf informatiepulsen en een stoppuls naar de "statisizer". De bedoeling van boven gegeven arrangement is, om de statisizer slechts via het toetsenbord te vullen, als op het moment van de startpuls "Acflop toetsenbord = 1" is, terwijl deze vulling zorgt, dat "Acflop toetsenbord := 0" uitgevoerd is, voordat de volgende startpuls zou kunnen komen. Het autonoom transport probeert de inhoud van de statisizer naar de vaste geheugenplaats te transporteren, zodra dit gelukt is, wordt dit door "Inflop toetsenbord := 1" naar de computer gesignaleerd.

Het ruwe schema, dat een karakter van het toetsenbord accepteert bestaat uit

"P(Inflop toetsenbord);
karakter:= toetsenbord;
V(Acflop toetsenbord)" .

In het kader van het tandenpoetsen moet de machine dus een keer "V(Acflop toetsenbord)" uitgevoerd hebben om voor deze invoer ontvankelijk te kunnen zijn.

1.2.2. Toetsenbord zonder expliciete "Acflop"

Bij het vullen van de statisizer kan men "Acflop toetsenbord:= 0" uitvoeren bij de afsluitende stoppuls; omdat het autonoom transport practisch instantaan zal plaatsvinden, wordt practisch tegelijkertijd "Inflop toetsenbord:= 1" uitgevoerd.

Omgekeerd: de luistervergunning van de "Inflop toetsenbord" zal slechts = 1 zijn, als er een toetsenbordprogramma daadwerkelijk in zijn operatie

"P(Inflop toetsenbord)"

is blijven hangen.

Als de ingreep effect heeft en dit programma weer in de coordinator in de categorie van de (gedeblokkeerde) onderbroken programma's wordt opgenomen -dwz. als de coordinator beslist, dat dit programma zijn P-operatie voltooid heeft- dan wordt "Inflop toetsenbord:= 0 uitgevoerd (en tevens "luisterbit toetsenbord:= 0", wanneer de wachtketen aan de Inflop toetsenbord hiermee leeg is geworden).

Aannemend, dat dit nu gedeblokkeerde toetsenbordprogramma nog aan de beurt komt -en daar is alles voor te zeggen om er in de coordinator via een prioriteitsregel voor te zorgen, dat dit inderdaad het geval zal zijn- dan zal het tweetal:

"karakter:= toetsenbord;
V(Acflop toetsenbord)"

snel daarna uitgevoerd worden.

De opmerking nu is, dat hieruit volgt, dat deze Acflop en Inflop "bijna altijd" elkaars inverse zullen zijn en dat men zich daarom kan afvragen, of de Acflop toetsenbord -en daarmee de operatie "V(Acflop toetsenbord)"- niet helemaal vervallen kan. We moeten dan wel opnieuw afspreken, wat er gebeurt als men in een te hoog tempo op de toetsen drukt.

In het voorstel 1.2.1. had het te snel drukken van de volgende toets geen effect. Nu is het voorstel, om het te snel indrukken van de volgende toets het vorige karakter te laten invalideren. Bv. als volgt.

Het indrukken van een toets heeft altijd ten gevolge, dat een karakter naar de statisizer gezonden wordt. Als dit echter gebeurt op een ogenblik, dat het vorige autonoom transport nog niet heeft plaatsgevonden (kans nihil) of als het autonoom transport plaatsvindt, terwijl "Inflop toetsenbord = 1" is, dan wordt op de geheugenplaats c.n. "toetsenbord" een non-valide karakter aangeboden (bv. numeriek groter dan 31). Hier kan het toetsenbordprogramma dan op testen.

2.Overwegingen bij de vergelijking.

Bij de vergelijking van deze twee oplossingen laat ik de financiele consequentie van een en ander buiten beschouwing: ik ben niet competent om deze te beoordelen en bovendien kon het wel eens niet zo verschrikkelijk veel uitmaken.

Voorts dienen we te bedenken:

a) dat de kans op "te snel de volgende toets indrukken" naar wij van ganserharte hopen, klein zal zijn;
b) dat het geen ramp is, als de boodschap in eerste aanleg verkeerd is doorgekomen, daar de operateur toch eerst zal afwachten of de "body" van de boodschap correct is uitgetypt, voordat hij de OK-melding geeft.

Anderzijds dienen we te [illegible] bedenken:

c) dat de kans op non-validiteit, omdat het autonoom transport nog niet heeft plaatsgevonden (1.2.2.) heel klein is, maar dat het haastig is om te eisen, dat de reactie op de ingreep onder alles omstandigheden "rap" zal zijn. (We hebben nog nergens essentiele haastsituaties en ik zou me graag de [illegible] vrijheid voorbehouden, om op gezette tijden -zeg eens in de vijf minuten- de machine een seconde doof te maken voor pak weg herindeling van de informatie op de trommel.) Als dan de bedienende [illegible] operateur met een kans van 1 op 300 "in de war" zou raken, zou dat toch wel hinderlijk zijn.

d) dat het gewenst is, dat in de coordinator de behandeling van het toetsenbordprogramma zich in zo weinig mogelijk van die van de andere programma's onderscheidt;

e) dat het gewenst is, dat de operateur, in twijfel of hij een toets heeft ingedrukt, zo gauw en zo zeker mogelijk weet, waar hij aan toe is.

Ik ga er van uit, dat opnamen van via het toetsenbord ingevoerde karakters en het uitprinten van deze karakters twee karaktersgewijze synchrone processen zullen zijn.

## 3. Een conventie voor teleprinterreservering.

Als de wachtketting van een hardware seinpaal leeg is -als er geen enkel programma op deze seinpaal wacht- dan zet de coordinator de bijbehorende luisterbit = 0. Deze wordt slechts = 1, als wel een programma op het positief worden van de seinpaal staat te wachten. Dit impliceert, dat ik me ingrepen kan besparen, door zo mogelijk er naar te streven, dat P-operaties op hardware seinpalen zo laat mogelijk gepasseerd worden: is de ingreep op dat ogenblik al binnen, dan [illegible] kunnen we dat meteen verifieren.

Hieruit volgt als algemene strategie, dat we de passering van een hardware seinpaal in de programma's zo ver mogelijk naar achteren schuiven. Wat aan conventies impliceert, zullen we zo dadelijk zien. (N.B. Ik weet, dat teleprinter-ingrepen zo weinig frequent voorkomen, dat het uitsparen van ingrepen op [illegible] efficiency-overwegingen hier een zwak argument is; ik wil echter de hele behandeling van de in-en uitvoer-troep zo uniform mogelijk houden, anders verdrinken we in de chaos.)

Het is duidelijk, dat verschillende programma's van de teleprinter gebruik kunnen willen maken; het is kennelijk niet de bedoeling, dat karakters van verschillende bronnen krislkras door elkaar getypt worden: voor het uitprinten van een samenhangende tekst moet een programma de teleprinter dus voor zich reserveren. We voeren hiertoe in de geprogrammeerde seinpaal "teleprinter vrij"; dit is een 2-waardige seinpaal.

Het synchroon met de teleprinter aanbieden van een stuk samenhangende tekst kan nu door twee soorten programma gebeuren -met weglating van alle hier niet irrelevante tellingen-:

```
type A:     " P(teleprinter vrij);
          L: P(Inflop teleprinter);
             teleprinter:= volgend karakter;
             V(Acflop teleprinter);
             if B then goto L;
             V(teleprinter vrij)   "     of

type B:     " P(teleprinter vrij);
          L: teleprinter:= volgend karakter;
             V(Acflop teleprinter);
             P(Inflop teleprinter);
             if B then  goto L;
             V(teleprinter vrij)    " .
```

In het eerste geval wordt de teleprinter vrijgegeven na de laatste tikopdracht, in het tweede geval pas na de terugmelding, dat het volgende karakter alweer getikt kan worden. Het is duidelijk, dat wij slechts een organisatie kunnen opbouwen, mits we hier een duidelijke keuzen doen; mijn voorkeur gaat op grond van de eerder gegeven overwegingen uit naar type A. (Deze voorkeur is niet uiterst dwingend; dit is een reden te meer om deze conventie in alle klaarheid en met alle nadruk te formuleren!)

4. Uittikken synchroon met het toetsenbord.

In het volgende veronderstel ik de operatie "P(teleprinter vrij)" uitgevoerd- waarover later meer- zodat niet, terwijl ik aan het "intikken" ben, een machineprogramma me de teleprinter onder de vingers weg kan kapen.

4.1. Met expliciete Acflop voor het toetsenbord.

4.1.1. De meest rigoureuze sluis, die ik me voor kan stellen bestaat uit de volgende cyclus:

```
"L: P(Inflop teleprinter);
    V(Acflop toetsenbord);
    P(Inflop toetsenbord);
    karakter:= toetsenbord;
    teleprinter:= karakter;
    V(Acflop teleprinter);
    if B then goto L"
```

Hier vraagt de machine pas om een volgend karakter van het toetsenbord, als de terugmelding van het typen van het vorige karakter binnen is. Hier weet de operateur drommels goed, waar hij aan toe is, als hij midden in het tikken van een boodschap -doordat iemand binnenkomt of door doofheid van de machine- gestoord wordt. Als de getikte letters van de boodschap alle nog correct zijn, maar hij weet niet precies meer wat hij heeft aangeslagen, dan zijn er drie mogelijkheden. (Ik neem, evenals bij de volgende toetsenbordprogrammatjes aan, dat zij in doofheid uitgevoerd zullen worden!)

a) het programma is (in de coordinator) in de eerste P-operatie blijven hangen, hoewel de bijbehorende ingreep al binnen is (de printer staat immers stil);
b) het programma is in de tweede P-operatie blijven hangen, omdat de ingreep er nog niet is -de operateur moet de volgende toets aanslaan
c) het programma is in de tweede P-operatie (in de coordinator) blijven hangen: het volgende karakter is via de statisizer al tot de machine doorgedrongen.

In alle drie de gevallen kan de operateur het volgende karakter aanslaan; in geval a) heeft het geen effect, in geval b) heeft het het gewenste effect en in geval c heeft het geen effect (niet het ongewenste). Alleen in het geval c zit er een klein, klein lekje, nl. als de operatie "P(Inflop teleprinter)" voltooid kan worden voordat het karakter goed en wel op het papier staat. Het karakter kan dan twee keer getikt worden.

4.1.2. Een iets minder rigoureuze sluis is:

```
"L: P(Inflop toetsenbord, Inflop teleprinter);
    karakter:= toetsenbord; teleprinter:= karakter;
    V(Acflop teleprinter, Acflop toetsenbord);
    if B then goto L"
```

Hier kan de machine alleen in de P-operatie blijven hangen; als de teleprinter stil staat, is dus "Inflop teleprinter" geen belemmering. Of (a) Inflop toetsenbord is de belemmering of (b) er is geen belemmering, maar de coordinator gunt de machine niet. Als nu weer de tekst voorzover uitgetikt correct is, dan kan de operateur maar een ding doen: het volgende karakter aanslaan en wachten.

In geval (a) is dit normaal het volgende karakter, in geval (b) was dit karakter al binnen en wordt zijn tweede aanslag genegeerd. Voordeel van dit arrangement is, dat 1 keer volgend karakte aanslaan beslist voldoende is. Ook dit sluisje heeft (in geval b) een lekje: vlak voordat hij de toets hernieuwd aanslaat, beeindigt de machine de P-operatie en maakt zij via de V-operatie de statisizer opnieuw ontvankelijk en de letter "X" verschijnt dus twee maal op papier. Het lekje is iets groter, omdat het niet te verkleinen is door de ingreep van het toetsenbord te vertragen. De kans, dat doorgaan van het toetsenbordprogramma en indrukken van de volgende toets net voldoende coincideren, lijkt me zo klein, dat we dit lekje wel kunnen accepteren. Een voordeel van dit arrangement is, dat in geval van twijfel nog 1 keer de volgende toets indrukken in elk geval op den duur soulaas moet brengen.

Nadeel van dit arrangement is, dat "te snel inslaan" hoegenaamd niet tot de machine doordringt en dat snel inslaan van "a b c" dus "a c" op het papier zou kunnen doen verschijnen. Ik weeg dit bezwaar niet zo heel erg en wel op grond van de volgende overweging: als "a c" op het papier verschenen is, dan is dat naar voor de operateur, want hij moet het correctieritueel uitvoeren. Als hem dit een paar keer is overkomen, zal hij er verder wel voor oppassen, dat dit hem niet te vaak overkomt: en de gedragslijn, die hij daartoe moet volgen, is duidelijk!

## 4.2. Zonder expliciete Acflop voor het toetsenbord.

We beschouwen nu de analoge versie van het programma uit 4.1.2.:

```
"L1: P(Inflop toetsenbord, Inflop teleprinter);
 L2: karakter:= toetsenbord;
     if  31 < karakter  then
                 begin  P(Inflop toetsenbord); goto L2 end;
     teleprinter:= karakter;
     V(Acflop teleprinter);
     if B then  goto L1"
```

Wat betreft het te snel aanslaan van de operateur is dit beter. Nemen we weer het te snel aanslaan van "a b c" in die zin, dat b aangeslagen wordt, voordat a correct is verwerkt. In 4.1.2. was de fout van de operateur, dat hij "c" aansloeg, voordat "b" getikt was; in dit geval moet hij, om het nu spaak te laten lopen "c" al slaan, voordat de "a" getikt is: de "a" wordt nl. door de te snel gekomen "b" bedorben tot een karakter, dat groter dan 31 is.

Laten we nu onderzoeken, hoe precies de operateur weet, waar hij aan toe is, als hij aarzelt wat hij heeft aangeslagen of wat door de machine geweigerd is. Het interessante geval is natuurlijk weer, dat de operateur zich even bedenkt en ziet, dat datgene, wat op papier staat, nog correct is. Het enige, wat hij kan doen is na enige bedenktijd vol goede moed het volgende karakte aanslaan, dat op papier verschijnen moet.

Stond het programma op "Inflop toetsenbord" te wachten, dan was deze actie goed. Nu het geval, dat dat volgende karakter al aanwezig was.

Was "Inflop toetsenbord = 1", maar was de machine te lui met haar reactie

hierop, dan vernietigt de nieuwe aanslag het al aanwezige volgkarakter en moet dus nettertijd nogmaals op de toets gedrukt worden.

Was inmiddels "Inflop toetsenbord:= 0" voltooid, maar had de coordinator het toetsenbordprogramma nog niet verder laten lopen, dan zal de operateur merken, dat het volgende karakter twee keer getikt wordt.

De kans hierop is te verkleinen, door de coordinator zo te construeren, dat, wanneer de coordinator de operatie P(Inflop toetsenbord) voltooit, dwz. "Inflop toetsenbord:= 0" uitvoert, het nu toegestane toetsenbordprogramma inderdaad zo snel mogelijk voortgezet wordt. Dit is de prijs, die we betalen voor het dubbele gebruik van "Inflop toetsenbord" en ik vind het niet mooi.

5. Een mengvorm.

Salvo errore et omissione combineert het volgende voorstel de deugden van beide systemen.

Elke keer, dat een toets wordt ingedrukt, wordt een karakter naar de staticiser gezonden; als op het moment van de startpuls de staticiser nog vol mocht zijn, dan neemt de staticiser een non-valide karakter op. (Kans nihil, dit is het zelfde loffelijke perfectionisme als in 1.2.2.)

De inhoud van de staticiser wordt nu in principe naar het geheugen doorgezonden per autonoom woordtransport, afhankelijk echter van de waarde van Acflop toetsenbord en Inflop toetsenbord op dat ogenblik. (De actie van Acflop toetsenbord is dus van het vullen van de staticiser verschoven naar het autonoom transport.)

We onderscheiden nu drie gevallen (het vierde kan zich niet voordoen).

a) Acflop toetsenbord = 1 en Inflop toetsenbord = 0.
Het karakter wordt zoals het zich in de staticiser bevindt, naar het geheugen doorgezonden; Acflop toetsenbord:= 0; Inflop toetsenbord:= 1;
(Dit als ondeelbare handeling. Dit is het normale geval van karaktertransport.)

b) Acflop toetsenbord = 0 en Inflop toetsenbord = 1.
Het karakter wordt in "non-valide vorm" in het geheugen neergezet.
(Dit is het geval, dat de tweede toets ingedrukt wordt, terwijl de machine nog niet gereageerd heeft op de ingreep van de vorige. We kunnen dan het vorige karakter nog met "non-valide" overschrijven.)

c) Acflop toetsenbord = 0 en Inflop toetsenbord = 0.
Het autonoom transport wordt onderdrukt en het karakter raakt "onder tafel".
(Op de ingreep is gereageerd en het is dus te laat, om het vorige karakter nog te herroepen; anderzijds is het tweede karakter wel te vroeg, omdat de machine er nog niet om gevraagd heeft.)

De vierde mogelijkheid komt niet voor, omdat de machine slechts Acflop = 1 zal zetten, als de Inflop = 0 is.

De structuur van het programma is nu als volgt:

```
"L1: P(Inflop toetsenbord, Inflop teleprinter);
 L2: karakter:= toetsenbord;
     if 31 < karakter then  begin V(Acflop toetsenbord); P(Inflop toetsenbord);
                                 goto L2  end;
     teleprinter:= karakter;
     V(Acflop teleprinter, Acflop toetsenbord);
    if  B then  goto L1"
```

Wie nu te snel achter elkaar "a b" aanslaat, zal merken, dat er of niets (heel snel) of "a" (iets minder te snel) getypt wordt. In beide gevallen moet hij het eerste ontbrekende karakter aanslaan.

Als nu de operateur aarzelt over wat er gebeurd is, maar de boodschap is, voorzover hij op het papier verschenen is, nog correct, dan kan de operateur na enige bedenktijd alleen maar vol goede moed het ontbrekende karakter (nogmaals?) aanslaan.

De teleprinter staat stil en "inflop teleprinter" kan dus geen belemmering zijn. We moeten nu drie gevallen onderscheiden.

a) Als Inflop toetsenbord de belemmering in de P-operatie is, dan wacht het programma op het volgende karakter en aanslaan ervan is dus correct.
b) Als Inflop toetsenbord = 1 is, maar de machine is wegens doofheid wat lui met de reactie op deze ingreep, dan maakt de extra aanslag het aanwezige karakter non-valide en is er dus ook niets kapot. (Na verloop van tijd raakt het programma dus in toestand a), dat het karakter nogmaals aangeslagen moet worden.)
c) Als "Inflop toetsenbord:= 0" door de coordinator voltooid is, maar het toetsen-bordprogramma is nog niet aan bod gekomen, dan is het volgende karakter inmiddels onherroepbaar tot het geheugen doorgedrongen. De extra aanslag -de herhaling- deert nu niet, want deze aanslag raakt onder tafel.

Het enige lekje, dat er nu nog in zit is hetzelfde als in het programma 4.1.2.. Als de operateur een inmiddels ingevoerd (en geaccepteerd) maar nog niet uitgetikt karakter uit ongeduld nog eens aanslaat, precies op het ogenblik, dat de machine de V-operatie uitgevoerd heeft, dan wordt het karakter twee keer ingevoerd. Dit is echter het risico, dat ik geneigd ben om te accepteren.

Aan de problemen van teleprinter-reservering en conventies voor herroepen van ingetikte boodschappen, dan wel OK-melding wilde ik een andere notitie wijden.

PS. Voor de misspelling van "staticiser" in het begin van deze notitie des schrijvers verontschuldigingen.