

Over de sequentiele interpretatie van een ponsband.HEP

Als fundamentele bouwsteen beschouw ik de integer procedure, die ik HEP noem. Bij het lezen van n-gats band levert hij in eerste instantie een van 2^n mogelijke waarden af, nl van en met 0 (voor blank tape) tot en met $2^n - 1$ (voor all holes) en wel steeds de waarde die overeenkomt met de volgende n-ade.

Hoe we aan de band komen is een onderdeel van de initialisering van HEP, waarover ik me liever niet wilde uitlaten. Ik neem dus aan, dat we weten, over welke band we het hebben. Die kunnen we dan n-ade voor n-ade aftasten: die band kan dan wel opraken!

De vraag rijst direct, hoe HEP moet reageren, als "einde band" gedetecteerd wordt.

Mijn voorstel is om het arsenaal van mogelijke, door HEP af te leveren waarden uit te breiden met een speciale waarde (ik denk aan 511 of -1), die "EINDE BAND" zal betekenen. Het is dan aan het programma om vast te stellen, of het nu genoeg gelezen heeft; zo nee, dat zal HEP opnieuw geïnitieerd moeten worden (band aanvragen).

Onmiddellijk rijst de vraag, hoe HEP reageert, als nadat "EINDE BAND" als waarde is afgeleverd, HEP weer wordt aangeroepen, voordat de volgende initialisatie heeft plaatsgehad.

Nu kunnen we drie kanten uit.

- 1) we kunnen dit interpreteren als lokaal deraillement van het programma
- 2) we kunnen dit interpreteren als de indicatie, dat er dus blijkbaar meer informatie gevraagd wordt en als onderdeel van HEP' impliciet de initierings-ritten plagen (om dieuwe band vragen etc.)
- 3) we kunnen weer "EINDE BAND" afleveren.

Omdat ik HEP beschouw als mijn meest fundamentele bouwsteen prefereer ik het laatste; het maakt het gebruik van HEP mogelijk zonder na elke aanroep te inspecteren, of "EINDE BAND" soms voorkwam "X:= HEP * HEP" kan nu. Een band wordt nu een halfrechte, oneindig lang in de toekomst, vanaf een zeker moment komen er alleen maar ~~XEINDEBANDXXXXXXXXXXXX~~ "EINDE BAND"'s op voor.

De routine HEP is van toepassing, als we als band willen verwerken, bv. de output van een registrerend meetapparaat. De rest van deze notitie beschouwt de inhoud van de ponsband als derivaat van een paginabeeld: hier ligt een code aan ten grondslag, bv. telexcode of FLEXOWRITER code.

CHAR

In het volgende zal ik een en ander toelichten aan de hand van Flexowriter code; voor telexcode is een analoge beschouwing te houden, zolang er geen losse carriage returns zonder line feed gegeven worden.

En ik moet me zelfs beperken tot een vrij speciale Flexowriter: een Flexowriter met vaste spatiering, zonder backspace waarbij de tabulatorstoppen op vaste posities staan. (En dit apparaat zullen wij zelfs ~~M~~ nog iets moeten idealiseren: over de minimum afstand, waarin de tabulatie "het nog haalt" , bv. 1 niet, twee wel, moeten we een vaste afspraak maken.)

CHAR nu is een routine, die zich interesseert voor het paginabeeld: het is de functie van CHAR om bij het aftasten van een band een verslaglegging te doen, die een-eenduidig is toegevoegd aan het met de band corresponderende paginabeeld.

- Ik stel me bij de definitie van het paginabeeld op het standpunt, dat
- 1) meervoudige onderlining niet van enkelvoudige onderlining is te onderscheiden
 - 2) ~~XXX~~ meervoudige "stroking" niet van enkelvoudige "stroking" is te onderscheiden
 - 3) dat de op de wagen ingestelde afstand van de regelopvoer niet tot het paginabeeld behoort, een extra regel blank wel
 - 4) dat op het papier de plaats van de linker kantlijn bekend is (wie elke regel begint met een extra spatie fikt een ander paginabeeld)

Ik stel me voor, dat CHAR bij herhaald aanroepen dit verslag zal afleggen, daarbij ~~M~~ het paginabeeld regel voor regel verslaand (in volgorde van ~~XXXX~~ boven naar beneden) en elke regel op zichzelf van links naar rechts.

Over de wijze van verslaglegging door CHAR zullen we enige conventies moeten treffen.

Ik stel me voor, dat CHAR niet elke regel met spaties tot een vaste maximum lengte hoeft aan te vullen, maar in plaats daarvan NLCR wel tot de output van CHAR te laten behoren; CHAR heeft dan wel de plicht om eventuele TABS en SPATIES aan het einde van de regel te onderdrukken.

Voor de verwerking van de TAB's stel ik voor ze te vertalen in een equivalent aantal spaties: op deze wijze geeft een reeks CHAR-waarden zonder NLCR een beeld, dat invariant is voor zijn plaats op de regel. Wel zal CHAR voor het bepalen van deze equivalentie de wagenpositie moeten bijhouden.

Wat betreft de onderlining en de stroking kunnen we verschillende wegen bewandelen.

- 1) elke "zichtbare, transporterende ponsing" bestaat in vier versies, al of niet met onderlining, al of niet met stroking
- 2) een aanroep van CHAR kan wel "underlining" en "stroking" separaat doormelden. In dit geval moet, om der wille van de eenduidigheid een vaste afspraak gemaakt worden over de volgorde, waarin deze ~~afsp~~ door CHAR worden doorgemeld (bv. eerst de onderlining, dan de stroke en dan het transporterende karakter).

Omdat CHAR dan toch verder moet kijken dan zijn neus lang is, preferer ik de eerste mogelijkheid. Deze heeft de voordelen, dat

- 1) we van de willekeurige afspraak over de volgorde verlost zijn
- 2) nu met elke aanroep van CHAR een positieverplaatsing ~~XXXX~~ van de wagen correspondeert.

CHAR heeft de eigenschap, dat mist de code bekend is, elke produceerbare band gelezen kan worden en over elke produceerbare pagina een uniek verslag wordt afgelegd.

Een willekeurige Flexowriterpagina is niet als willekeurige Telexpagina te interpreteren, noch omgekeerd. Ik stel daarom voor, om de codeonafhankelijke CHAR ~~XXXXXXXXXX~~ af te schaffen en zoveel CHAR's in te voeren, als we codes (i.e. apparaten)

