

Kennismaking met de computer

B. Koers

R.U. Groningen

Summary

How to start computer-science or informatics at secondary education level? Nobody seems to know, at least not in the Netherlands. This article tells about (another) experiment that gives students a first confrontation with the computer. The student should get an impression of what the computer is, and can do for him. Of course within a limited time. This experiment focusses on word-processing, working with high-level-commands and therefore needs a 'real' computer, and not a micro. Students of secondary and university level appreciated the short course very much.

In dit artikel wordt uiteengezet hoe met het vak informatica (computerkunde) op de middelbare school begonnen kan worden. Centraal staat *de eerste kennismaking* van de leerlingen met de mogelijkheden van de computer.

De kennismaking van leerlingen met een nieuw vak kan op vele manieren geschieden. Wat gebeurt er in de eerste schoolweek bij biologie, economie of wiskunde? Wordt er begonnen met een algemeen overzicht of laat men toepassingen zien? Begint de leraar op pagina 1 van het boek of geeft hij een eigen introductie? Waarschijnlijk kiest iedereen zijn eigen methode, die per vak verschillend zal zijn.

Wat betreft informatica hebben we gekozen voor:

De leerling dient binnen kort tijdsbestek, op grond van zoveel mogelijk zelfstandig werken, een indruk te krijgen van wat een computer is en kan.

Deze doelstelling houdt onmiddellijk in, dat we een introductie, waarbij eerst veel aandacht wordt besteed aan het leren van een programmeertaal (ECOL, BASIC) volstrekt afwijzen. Helaas zijn echter veel bestaande vormen van computeronderwijs hierop gebaseerd. Bij dat onderwijs duurt het vaak bijzonder lang voordat de leerling voor zichzelf interessante toepassingen kan maken. Het duurt vele weken voordat de leerling de gebruikte programmeertaal kent en eigen initiatief kan ontplooien. Dit is vooral in het begin vervelend en weinig motiverend.

Een alternatief denken we gevonden te hebben door het ontwikkelen van een computer-practicum, waarbij de leerlingen na een zeer korte inleiding zelfstandig met de computer aan het werk kunnen. Met behulp

van krachtige commando's, zoals *sorteer*, *zoek*, *vervang* (zgn. hoog-niveau-commando's) kan men binnen enkele uren zeer interessante opdrachten uitvoeren.

In het door ons ontworpen practicum krijgt de leerling direct ongeveer 15 commando's ter beschikking. Wat hij moet doen is aan de hand van een practicum-handleiding de functies van de diverse commando's bestuderen om er vervolgens problemen mee op te lossen. Hierbij gaat het ten eerste om het kiezen van de juiste commando's en ten tweede om het in de goede volgorde gebruiken daarvan. Aan de opgaven 14 t/m 18 (zie later) is te zien tot welke resultaten leerlingen binnen vier uur kunnen komen.

Het beschreven practicum is het resultaat van een project, dat twee jaar geleden in samenwerking met een aantal leraren HAVO/VWO gestart is door de afdelingen Informatica en Didaktiek van het Mathematisch Instituut van de Rijksuniversiteit Groningen. Het project staat onder leiding van de medewerkers John van Meurs (informatica) en Jan Sloff (didaktiek). De schrijver van dit artikel nam als student-assistent deel aan het project en zorgde voor de praktische begeleiding en de documentatie.

Faciliteiten

Tot nu toe hebben ongeveer 200 leerlingen uit HAVO 2-5 en VWO 2-6 deelgenomen aan experimenten met ons practicum.

Zij hebben in hun vrije tijd gewerkt op een PDP11 mini-computer van de afdeling Informatica, werkend onder het beheerssysteem UNIX. Afhankelijk van de mogelijkheden, die de schooldirecties boden, konden de meewerkende leraren kiezen uit twee mogelijkheden:

- a. Er wordt gedurende enkele weken een terminal van de afdeling op school geplaatst, zodat via een normale telefoonlijn (en kosten) met de computer gewerkt kan worden.
- b. Men komt met groepen leerlingen een of meerdere middagen naar de universiteit.

In geval b) zijn de faciliteiten tot nu toe zodanig dat maximaal 32 personen tegelijkertijd met de computer kunnen werken.

Behalve voor middelbare scholen wordt het practicum ook gebruikt tijdens de introductie van eerstejaars studenten wiskunde en informatica aan de Rijksuniversiteit Groningen.

Onze conclusie is dat leerlingen en studenten erg enthousiast zijn, eigen initiatief kwijt kunnen en wezenlijke zaken uit de informatica leren kennen. Alvorens leerlingreacties en leerlingwerk te beschrijven, wordt een samenvatting van de inhoud van het practicum gegeven.

Practicum

Het practicum bestaat uit drie delen:

- a. theoretische voorbereiding
- b. werken met hoog-niveau-commando's
- c. tekstverwerking met een editor

a. theoretische voorbereiding

In de theoretische inleiding wordt verteld wat men tenminste moet weten voor men met de computer kan gaan werken. Uitgelegd moet worden hoe een computersysteem(pje) is opgebouwd, met name terminologie als beeldscherm, toetsenbord, terminal, geheugen, programma en printer. In het bijzonder moet aandacht geschonken worden aan het begrip *file*.

Een *file* is een hoeveelheid, in het geheugen opgeslagen informatie, die bekend is onder een bepaalde naam, de *filenaam*. De inhoud van een file kan heel verschillend zijn, bijvoorbeeld een brief, een gedicht, een verslag, een serie meetgegevens, een rij getallen, kortom: een verzameling lettertekens. Onder lettertekens worden alle tekens van het toetsenbord verstaan. Het uitleggen van het begrip file en het uitleggen van de bediening van de apparatuur is voldoende om met het praktisch werk te beginnen. In totaal hoeft dit niet meer dan 1 lesuur te kosten. Parallel aan het practicum kan eventueel een uitgebreidere inleiding gegeven worden.

b. werken met hoog-niveau-commando's

De leerlingen krijgen een lijst met 15 commando's en een zeer beknopte omschrijving van hun functies. Daarnaast krijgt men de practicum-handleiding, waarin deze commando's geleidelijk worden geïntro-

duceerd. Er wordt begonnen met het leren maken van nieuwe files. Hieronder volgt de tekst van de practicum-onderdelen "Het maken van files" en "Het maken van programma's". Een programma is hier een 'samengesteld commando', d.w.z. een combinatie van commando's die achterelkaar uitgevoerd worden. Programmeren is het kiezen van de juiste commando's en het in de juiste volgorde plaatsen daarvan. In feite verschilt het programmeren m.b.v. hoog-niveau-commando's niet van het programmeren in een echte programmeertaal. Men kan zeggen dat de hoog-niveau-commando's een eigen taal vormen, die zeer klein en gemakkelijk aan te leren is.

In het reeds genoemde beheerssysteem UNIX zijn de meeste commando's standaard aanwezig, terwijl niet aanwezige commando's eenvoudig gemaakt kunnen worden. De file-structuur van UNIX maakt het mogelijk dat alle groepen leerlingen de beschikking hebben over een eigen file-bibliotheek. In het onderstaande volgt een deel van de practicum-tekst, die de leerlingen in groepjes van twee moeten doorwerken. Hierbij blijkt weinig begeleiding nodig te zijn.

Het maken van files

Wanneer je de onderstaande tekst

De slang
is niet heel dik, maar naar proporties te lang
schoon sommige slangen, die van Amerika komen
bijna zo dik zijn als eikebomen
en menige Leidse diender maakt een mens niet half
zo bang
als zulk een Amerikaanse slang.

wilt opslaan in het geheugen van de computer zul je via commando's moeten opgeven dat je een file wilt maken. Daar er in het geheugen veel meer files zitten, zul je deze file een naam moeten geven, bv. de naam 'slang'. De procedure is nu als volgt; achter het commando-teken (%) tik je in:

maak slang

en druk vervolgens de RETURN-toets in. De computer verwacht nu een tekst en zal deze opslaan in de file 'slang'. Je gaat vervolgens de hele tekst intypen; aan het eind van iedere regel druk je de RETURN-toets in en je ziet dan dat de cursor (wit rechthoekje, dat aangeeft waar je bent op het scherm) naar de volgende regel verspringt. Alles wat je intikt wordt als tekst beschouwd, dus ook eventuele commando's. Het enige commando dat niet als tekst wordt beschouwd is:

CRTL D (tegelijk op CTRL en D drukken).

Dit commando geeft de computer het sein dat de tekst beëindigd is. We gaan nu meteen controleren of er geen fouten zijn gemaakt en of de file 'slang' wel echt in het geheugen zit. Allereerst volgt nu het commando om te zien welke files je ter beschikking hebt:

files gevolgd door RETURN.

Als je de inhoud van de file 'slang' wilt zien, dan kun je het commando

haal

gebruiken, maar dan moet je wel opgeven welke file je zien wilt:

haal slang (RETURN)

Wat er gebeurt is dat de file 'slang' opgehaald wordt en in één keer afgedrukt wordt. Als je hetzelfde met een erg lange file doet, vliegt de tekst over het scherm, terwijl je niets kunt lezen, behalve de laatste pagina van de file, die op het scherm blijft staan omdat het het laatste stuk is. Wil je een langere file helemaal bekijken dan is het commando *page* noodzakelijk: de tekst komt pagina voor pagina op het scherm. De volgende pagina verschijnt na een RETURN. In totaal dus:

haal slang | page (RETURN)

We hebben nu een 'samengesteld commando' gemaakt. Commando's kunnen gekoppeld worden door het koppel-teken |. Hierbij geldt dat de uitvoer na het eerste commando (*haal*) gebruikt wordt als invoer voor het tweede commando (*page*).

Opgaven

1. Voer de tekst *slang* in op de manier zoals boven beschreven. Controleer dat hij in het geheugen staat. Is gebruik van het commando 'page' noodzakelijk?
2. Bekijk welke files je ter beschikking hebt. Wanneer zijn ze gemaakt?

Het maken van programma's

We hebben nu gezien dat de computer een tekst kan opslaan, maar hij kan gelukkig nog meer. Het toestel is onder andere ook in staat om opgeslagen files te bewerken. Neem nu eens aan dat de onderstaande tekst op een file met de naam 'zin' staat.

De moderne talen zijn frans, Duits en engels.

Met het commando *woorden* rafel je de tekst uiteen tot losse woorden. Ga na dat het vereiste programma (een samengestelde instructie) luidt:

haal zin | woorden (RETURN)

Op je beeldscherm zie je nu verschijnen:

De
moderne
talen
zijn (enz)

We kunnen de tekst ook in letters uit elkaar halen i.p.v. in woorden. Dan wordt in verband met de lengte van de file wel het commando *page* noodzakelijk:

haal zin | letters | page (RETURN)

Met het commando *tel* worden de regels en de woorden van een file geteld, bij voorbeeld:

haal zin | tel (RETURN)

Opgaven

3. Tel het aantal regels van de file 'slang'. Tel tevens het aantal woorden.
4. Wat zal het resultaat zijn van: *haal slang | woorden | letters | page?* (Eerst voorspellen, dan doen.)
5. Tel het aantal letters in de file 'slang'. Idem voor 'keer'. Wat merk je?
6. We hebben al gezegd dat bij een koppel-teken de uitvoer van het vorige commando de invoer van het volgende commando is. Ga eerst voor jezelf na wat er in de volgende instructie gebeurt na elke koppeling. Controleer dit met de computer. Het programma luidt:

haal slang | letters | tel | tel (RETURN)

Het is ook mogelijk veranderingen in een file aan te brengen. Je kunt bijvoorbeeld in de file *slang* alle stukjes tekst 'tekst 1' vervangen door 'tekst 2'. (Zo'n stukje noemen we een *string*). Dit gebeurt d.m.v. het commando *vervang* tekst 1 tekst 2.

Tik het hiervoor benodigde programma in en kijk of het resultaat klopt.

Als je nu weer de file *slang* bekijkt zul je zien dat er in het geheugen niets veranderd is. Daar kun je weer gewoon mee verder werken. Zou je deze file wel willen veranderen, dan moet de uitvoer van je programma niet op het scherm komen, maar naar het geheugen gestuurd worden. Na afloop van het samengestelde commando moet dan een extra *maak*-opdracht gegeven worden voor een nieuwe file. Het proces gaat nu als volgt:

haal slang | vervang i j | maak newslang (RETURN)

Je kunt nu de nieuwe file ophalen en bekijken.

Opgaven

7. Vervang in de file *slang* alle dubbele a door dubbele o.
8. Vervang elke letter e door de letter a en daarna elke oorspronkelijke a door een e. Denk hier goed over na, roep niet te snel om hulp. Als aanwijzing geven we dat je de oorspronkelijke a's herkenbaar moet maken.
9. Je kunt nu leuke dingen doen:
vervang *slang* door muis
vervang *slang* door Piet
en wat je zelf maar wilt.

Tot zover de letterlijke tekst. In deel III van de tekst worden de overige commando's behandeld op gelijke wijze. We geven de korte omschrijving uit de commando-lijst.



keerom	Geeft van alle regels de leettertekens in omgekeerde volgorde.
roterlinks	Zet van alle regels het eerste woord achteraan. Het scheidingsteken tussen woorden is de spatie.
roterrechts	Zet van alle regels het laatste woord vooraan.
sorteer	Sorteert de regels van een tekst op alfabet.
uniek	Gooit dubbele regels eruit.
zoekstring \$	Geeft alle regels waar de tekst \$ in voorkomt. \$ is een vrij te kiezen rijtje leettertekens (characters).
zoekchar \$	Geeft alle regels waar een van de characters van de string \$ in voorkomt.

Nu kan men de volgende opgaven maken:

10. Vindt alle woorden in 'slang' waar de string 'men' in voorkomt.
11. Tel alle woorden in 'slang' waar een 'i' of een 'e' in voorkomt of beide. Tevens die waar een 'i' en een 'e' in voorkomen.
12. Sorteert een tekst op uitgang van woorden.
13. (Denk bij deze opgave eerst na over de vorm van de file.) Maak een lijst van 10 kennissen. Sorteert deze op achternaam. Sorteert ook op de laatste letter van de achternaam.

14. In het geheugen staat de file 'leden'. Dit is de administratie van een vereniging. De secretaris beschikt nl. over een computer en gebruikt deze voor zijn ledenbestand. Hij voert de gegevens als volgt in:
 - achternaam spatie voorletters spatie straatnaam spatie huisnummer spatie geboortear spatie streepje geboortemaand streepje geboortedag
 - a. Druk alle mensen af die jarig zijn in mei.
 - b. Sorteert deze op achternaam.
 - c. Tel het aantal mensen dat aan de Hooiweg woont.
 - d. Voor het bezorgen van het clubblad is het handig een lijst te hebben van alle leden, gesorteerd op straatnaam. Maak zo'n lijst.

Na introductie van extra opdrachten om files te vergelijken en aan elkaar te plakken volgen nog de opdrachten:

15. (vervolg van opgave 14)
 - De gemeente geeft jeugdsubsidie voor die leden, die jonger zijn dan 23 jaar. Hoeveel leden zijn dit? Geef een lijst.
16. Ga na of in een zelf te kiezen tekst palindromen voorkomen. Dit zijn woorden die zowel van achteren naar voren als van voren naar achteren gelezen kunnen worden. (bv. lepel)
17. Op de middelbare school wordt een proefwerk Engelse woordjes gehouden. De leraar gebruikt de computer om na te kijken en cijfers te geven.

Hij vraagt 10 woordjes, die staan op de file 'juist'.

a. Maak files met antwoordlijstjes van Piet, Tinke en Henk. Bepaal zelf de fouten die je ze laat maken.

b. De beoordeling is: per fout een punt eraf. Welke cijfers hebben de leerlingen?

18. Spelletje.

Maak een file 'raden'. Sla hierin een woord van 4 letters op, dat je practicum-genoot niet weet. Deze moet het woord raden door een woord van 4 letters in te typen en de computer te laten zeggen welke letters in het woord zitten. Maak een programma en test of het werkt.

Tot zover het werken met hoog-niveau-commando's.

c. tekstverwerking met een editor

Nu de leerling iets gezien heeft van de mogelijkheden van een computer, introduceren we het begrip *editor*. Een editor is een programma voor het systematisch veranderen van tekstfiles. Er kan gebruik gemaakt worden van een aantal basis-commando's, die aangeduid worden met een letter. Deze letter is een afkorting voor de opdracht die uitgevoerd wordt.

Wij beschrijven en gebruiken de bekende Edinburg Editor, die op vele computersystemen aanwezig is. Is deze er niet, dan zal een andere gebruikt moeten worden. De overgang vanuit hoog-niveau-programma's naar de editor geven we aan met het commando *fedin*, gevolgd door de editor-opdrachten. Als voorbeeld:

```
haal file ! fedin "f/stap/s/stop" ! sorteer ! page.
```

Een file wordt opgehaald, de string 'stap' wordt opgezocht (find) en vervangen door 'stop' (substitute), er wordt gesorteerd en afgedrukt. De werking van de editor berust op het bestaan van een *file-pointer*. Dit is een (denkbeeldige) wijzer in de file, die ergens tussen twee characters staat. De filepointer (fp) kan bestuurd worden met de editor-commando's. Als hij op de gewenste plaats staat, kan met andere commando's de tekst veranderd worden. We geven een korte beschrijving van enkele commando's.

f/.../	find text	zet de fp voor gewenste tekst
s/.../	substitute	vervang de met f gevonden tekst door een andere
i/.../	insert	plaats tekst links van de fp
v/.../	verify	test of tekst rechts van de fp staat
m	move	zet fp aan het begin van de volgende regel
m-	move back	zet fp aan het begin van de vorige regel
w	word	zet fp aan het begin van het volgende woord
w-	word back	zet fp aan het begin van het vorige woord
r	right	zet fp een positie naar rechts
l	left	zet fp een positie naar links
k	kill line	verwijder de huidige regel

e	erase	verwijder het karakter rechts van de fp
e-	erase back	verwijder het karakter links van de fp
p	print	druk de huidige regel af
pn	print times	druk de eerstvolgende n regels af

Verdere mogelijkheden:

1. Herhaald uitvoeren van een commando, door het gewenste aantal malen erachter te plaatsen. (0 voor zo vaak mogelijk).
2. Groepering van commando's, door plaatsing van hakenparen (...).
3. Keuze tussen commando's.
Bv. (c1, c2): als c1 faalt wordt c2 uitgevoerd.

Hiermee zijn belangrijke bouwstenen voor het programma geïntroduceerd. We zien opnieuw dat er niet een grote programmeer-taal geleerd hoeft te worden, maar dat de leerling a.h.w. programmeert in de editortaal. Deze taal is abstracter dan die der hoog-niveau-commando's, wat tot gevolg heeft dat er langer over de te maken opgaven nagedacht moet worden.

Enkele voorbeeld-opgaven:

1. Merk alle regels waar 'het' in voorkomt met een *
2. Druk de regels 5 t/m 10 af
3. Verwijder alle regels die niet met een 'a' of een 'e' beginnen
4. Schrijf het commando 'vervang' als editor-programma

Mogelijke oplossingen zijn:

1. "(f/het/10i/*m)0"
2. "m4p6" of "m4(pm)6"
3. "(v/a/m,v/e/m,k)0"
4. vervang aap noot: "(f/aap/s/noot)0"

Opgave 4 illustreert dat de editor-commando's van een lager niveau zijn dan de hoog-niveau-commando's. In feite zijn de hoog-niveau-commando's niets anders dan editor-programma's met een eigen naam. Het zal duidelijk zijn dat de hoog-niveau-taal nu gemakkelijk uitgebreid kan worden met zelfbedachte commando's.

Tenslotte zij vermeld, dat onderdeel c van het practicum even geleidelijk gepresenteerd wordt als onderdeel b. Er zijn vele voorbeelden gegeven en men maakt eerst korte opgaven, voordat men toekomt aan de voorbeeldopgaven 1 en 3. Wat betreft de tijdsduur: De meeste leerlingen hebben voldoende aan 4 uur per onderdeel.

Leerlingreacties -werk en -fouten

In de inleiding is al gewezen op het grote enthousiasme van de leerlingen. Waar een terminal op school was geplaatst, was deze in tussenuren en pauzes, voor en na de lessen continu bezet. Maar ook als een groep leerlingen naar de universiteit kwam, waar men tien

terminals ter beschikking had, viel het fanatieke werken op.

Nu is dit enthousiasme voor een deel te verklaren uit het lijfelijk contact met de computer. Men ziet onmiddellijk resultaat van wat men intikt, het is iets nieuws, men mag zelfstandig werken. We kennen dit ook van ECOL of BASIC cursussen. Toch constateren we bij ons practicum een heel andere werksfeer, doordat men meer bezig is met dingen die men zelf maakt. De meest uiteenlopende files verschijnen. We zien versjes, grapjes over leraren, liefdesbrieven of zomaar allerlei nonsens.

Juist doordat men bezig is met zaken uit de eigen belevingswereld, leert men haast ongemerkt geweldig veel. Dit speelse werken met eigen files gebeurt zowel in lagere als in hogere klassen. Bij de evaluatie van het practicum blijkt dat we deze twee groepen duidelijk moeten onderscheiden.

hoog-niveau-commando's

De hogere klassen blijken moeiteloos dit deel te doorlopen. In de lagere klassen komen we nogal wat problemen tegen, met name als er zelfstandig gedacht moet worden. De problemen zijn terug te voeren tot a) niet nauwkeurig lezen en b) moeite hebben de pas geleerde dingen in een volgende opgave toe te passen. De eerste opgaven worden moeiteloos gemaakt, nadat men gewend is aan het toetsenbord en filesysteem. Bij opgave 10, het vinden van de woorden waar de string 'men' in voorkomt, komen in klas 2 en 3 de problemen. Een oplossing die gegeven werd is:

haal slang | zoekstring men | page

Dit is fout, daar nu alle regels met 'men' erin afgedrukt worden, terwijl de bedoeling was alleen de woorden waar 'men' in voorkomt. De oorzaak van deze fout ligt in het feit dat dit de eerste opdracht is, waarbij men meerdere "nieuwe" commando's achter elkaar moet plaatsen. Nadat men hierop gewezen is levert de opgave weinig problemen meer op. Men komt snel tot de oplossing

haal slang | woorden | zoekstring men | page

Bij opgave 13 (zie boven) moet men bedenken, dat de definitie van 'sorteer' gevolgen heeft voor de filestructuur. De achternaam dient vooraan te staan op de manier van het telefoonboek. Als geschreven wordt 'van de Berg', dan vinden we deze naam na sorteren terug bij de v's, wat niet de bedoeling is. Het laatste deel van opgave 13 leverde nogal eens op:

haal naam | keerom | sorteer | page

Daar de computer direct laat zien dat deze oplossing niet correct is, kan de leerling snel doorgaan met het herstellen van de fout, zonder dat begeleiding van een docent nodig is.

Ook bij opgave 14, die zeer enthousiast gemaakt wordt, zien we dat vooral jonge leerlingen niet goed nadenken over wat het gevolg zal zijn van hun programma. Bij vrijwel alle ATH2 en HAVO2 leerlingen loopt de oplossing van 14a via een aantal fouten. Een voorbeeldregel uit de file 'leden' is:

jansen km groteweg 131 groningen 1950-5-14

Hier moet je als begeleider de leerlingen ertoe zetten bijzonder nauwkeurig te omschrijven waaraan ze zelf zien, dat iemand jarig is in mei. Uit het gesprek:

- ll: aan die vijf
lr: wat moet je dan intikken?
ll: haal leden | zoekstring 5 | page
lr: probeer maar
ll: klopt niet, want ook de geboortedatum 1955-2-12 zit er bij
lr: kijk nu eens goed naar de lijst van personen, die de computer je gegeven heeft. Valt je iets op?
ll: alles met een 5 erin geeft hij
lr: dat klopt met jouw opdracht, he?



Sommige vinden nu direct de oplossing van '-5-', anderen gaan nog via '-5', '5-' of wat erger, maar wel voorstelbaar is, 'mei'. Uit dit laatste spreekt een grenzeloos vertrouwen in de computer.

- ll: hij weet toch wel dat 'mei' de vijfde maand is? wat een stom ding.
lr: precies, hij doet alleen wat jij zegt

Al met al leert men vrij snel een tekstfile te beschouwen als een verzameling losse characters, die voor de computer niets met elkaar te maken hebben. Dit in tegenstelling tot de menselijke geest, die vrijwel automatisch bepaalde patronen herkent, zelfs onbewust. Het veranderen van de visie op tekstfiles is van groot belang bij het 'fedin' gedeelte. Ook de functie van de spatie als echt character is voor sommigen een openbaring.

editor-opdrachten

Dit deel blijkt voor tweedeklassers te moeilijk te zijn, vanwege het hogere abstractie-niveau. Voor hun gevoel stellen de letters niets meer voor en men raakt de kluts kwijt bij kreten als "(f/he/3wi/*/10)0". Onze conclusie is, dat dit deel op zichzelf niet te moeilijk hoeft te zijn, maar dat de huidige vormgeving niet geschikt is. Er moeten meer voorbeelden en korte oefenopgaven komen, zodat men langzaam went aan de abstracte notatie.

Dit wennen aan het abstractie-niveau komen we ook tegen bij de klassen 4, 5 en 6, maar hier is men toch vrij vlot over de beginproblemen heen. Men vindt de

fedin-opdrachten best interessant en later zelfs leuker dan de hoog-niveau-commando's, omdat er veel meer gepuzzeld (geprogrammeerd) moet worden. Probeer zelf de volgende opgave eens:

Druk (indien aanwezig) ieder vijfde woord van een regel af.

Via fouten komt men meestal wel op de goede oplossing. De fouten wijzen de leerlingen erop hoe nauwkeurig er gewerkt moet worden. Bv. bij de opdracht 'merk alle woorden "de" met een * ervoor en erachter' reageert een leerling met: "He, ook de "de"'s binnen een woord zijn gemerkt." (de kinderen hoorden).

Conclusie: niet de twee characters d en e zijn bepalend, maar het viertal 'spatie d e spatie'.

Een heel bijzondere fout is wel het terechtkomen in een eindeloze lus. Het fedin-achtervoegsel 0 geeft de mogelijkheid een aantal commando's oneindig vaak uit te laten voeren. Gelukkig vat UNIX de 0 op als maximaal 5000 maal, dus de schade blijft beperkt. Toch is het voor de leerling best vervelend, want 5000 maal vergt ook bij een computer wel enige tijd.

Een ander typisch informatica-probleem vormt het tellen. Bv. de filepointer staat in regel 6, ik wil naar regel 10, wat moet ik dan doen? m3?, m4?, m5? Ook hier geldt weer dat de computer zelf na een aantal pogingen het antwoord geeft, alhoewel het natuurlijk slimmer is de practicum-beschrijving goed te lezen. Steeds moet men zich afvragen waar de filepointer staat na elk onderdeel van het editor-programma. Als dit precies gebeurt zijn ook de moeilijker opgaven best te maken.

Verdere ontwikkeling

Dit project is voortgekomen uit de overtuiging van

onze afdeling dat introductie van informatica op de middelbare school noodzakelijk, maar niet eenvoudig is. De stelling "door leerlingen BASIC te leren, verkrijgen ze een verkeerde programmeer-techniek, die in het vervolgonderwijs zeer moeilijk af te leren is", is juist, maar daarmee zijn we niet klaar. Daarom is er in samenwerking met een aantal enthousiaste leraren gezocht naar alternatieven. Gedurende een winter werd door medewerkers van de afdeling Informatica om de veertien dagen 's avonds college gegeven. In de volgende winter is het hierboven beschreven onderwijs ontwikkeld, aan de leerlingen voorgelegd en geëvalueerd.

Het is de bedoeling, de geschetste introductie verder te ontwikkelen in de richting van onderwijs in een Algolachtige taal. Er is in enkele klassen al geëxperimenteerd met een aansluitend practicum Pascal programmeren. In de lagere klassen was de overgang nog te groot en denken we aan een tussenliggend practicum "Turtle Geometry", grafisch werken met een terminal. Leerlingen uit de hogere klassen bleken zich goed te kunnen redden met uitsluitend een boek over programmeren in Pascal, doordat men in het inleidend practicum de basisbouwstenen van het programmeren, nl. het sequentieel, herhaald en voorwaardelijk uitvoeren van commando's, al had leren kennen. Verwerking van de programma's verliep vlot, doordat de Pascal-programma's als gewone tekstfiles ingetikt en gewijzigd konden worden. Het kennen van een editor bleek hierbij van groot nut.

Getracht zal worden bij het programmeer-onderwijs in een Algolachtige taal aansluiting te vinden bij andere onderdelen van het middelbare schoolonderwijs, o. a. het natuurkunde practicum. Onzes inziens is de tot nu toe ontwikkelde introductie een goede basis voor verder onderwijs in de informatica.