



UCSD Pascal

Geïmplementeerd op de Apple

H. J. C. Otten

Het bekende UCSD Pascal operating system is nu ook op de Apple II geïmplementeerd. Ingenieursbureau Schröder uit Eindhoven heeft ons het Language system ter beschikking gesteld om Apple Pascal uit te proberen. Met UCSD Pascal verandert de Apple in een geheel andere machine met een volwassen operating system.

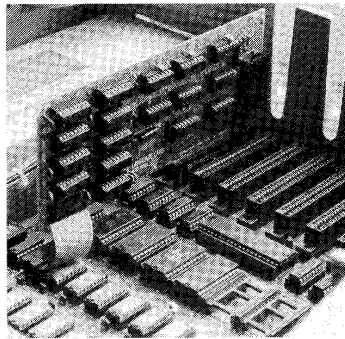
Systeemconfiguratie

Om UCSD Pascal op de Apple te kunnen draaien is een Apple II met 48K RAM en minimaal een floppy disk drive nodig.

Apple Pascal is een van de weinige UCSD Pascal implementaties die met één disk drive kunnen werken.

Het language system bevat:

- Language card met 16K RAM en de autostart monitor ROM.
- Nieuwe PROM's voor de disk interface.



- Diskettes met het UCSD Pascal operating system.
- Documentatie.

De language card (zie foto) die in plaats van de overbodig geworden Applesoft ROM card komt, bevat 16K dynamische RAM, waarmee het totaal op 64K dynamische RAM komt. Twee blokken van 4K zijn via memory banking aangesloten om de in/uit nog wat ruimte te geven.

De autostart monitor ROM geeft de Apple vooral voor de Basic interpreters een paar handige eigenschappen:

- De reset toets heeft niet een desastreus gevolg meer.
- Verbeterde screen-editing mogelijkheden.
- Autostart van de Apple.
- Page mode voor listings bijvoorbeeld.

Het merendeel van de oude monitor routines zijn op dezelfde adressen blijven staan.

UCSD Apple implementatie

Het UCSD (University of California, San Diego) Pascal is opgezet als een machine onafhankelijk operating system. Het gehele operating system is in Pascal geschreven, met uitzondering van de, machine afhankelijke, p-code interpreter. Bij het operating system is uitgegaan van een videoterminal als hoofd in/uit-apparaat. Tevens moet er een achtergrondgeheugen voor file opslag aanwezig zijn.

De mini floppy disk drive is als achtergrondgeheugen geschikt, waarbij wel een nieuwe indeling van de diskette ten opzichte van DOS Version 3.2 is gebruikt. Het gevolg is dan ook dat DOS V3.2 en Pascal files niet zijn



te combineren. De eigenschappen van de videoterminal van de Apple (40 karakters per regel) en de eisen van UCSD Pascal (80 karakters per regel) komen niet overeen. Met een truc is dit toch op een niet zo elegante wijze gerealiseerd. Een complete pagina tekst met 80 karakters per regel wordt in de Apple verdeeld in een linker en een rechter deel.

Door een druk op een toets is òf de rechter òf de linker pagina in beeld. Op de Pascal Apple kan nog steeds met Basic worden gewerkt. Op de Pascal Basic diskette staan Apple-soft, Integer Basic en DOS Version 3.2, die na laden in het geheugen de Apple weer normaal in Basic laten functioneren.

Operating system

Vanuit het operating system kunnen de diverse onderdelen van UCSD Pascal worden geactiveerd. Dit zijn de editor, de filehandler, de Pascal compiler, de linker, de library en de 6502 assembler. Ook kunnen gecompileerde programma's worden gestart. Het operating system en de onderdelen van het operating system melden zich met een prompt line waaraan we kunnen zien in welk onderdeel we zitten en welke commando's mogelijk zijn.

Editor

De uitgebreide editor van UCSD Pascal is sterk op een videoterminal gericht, en vereist cursor besturing en 80 karakters per regel. Het is een context editor die uiterst geschikt is om Pascal programma's te schrijven. Er zijn vele opties aanwezig en een aantal krachtige tekstverwerkingscommando's, zoals zoeken en veranderen van willekeurige strings en formatting (links en rechts justify). In de GOTOXY-procedure worden de specifieke eigenschappen van de videoterminal aangepast aan de eisen van de editor. Deze GOTOXY-procedure is te herschrijven en in het operating system op te nemen. Een voorbeeld hiervan ziet u in de lijst.

Pascal interpreter

Voor de Apple versie van UCSD Pascal was het nodig een nieuwe interpreter voor de p-code te schrijven in 6502 machinetaal en aan de Apple aan te passen. De gebruiker merkt niets van deze interpreter. Het is een stuk software dat zich net zo gedraagt als de microprocessor. Men

spreekt dan ook vaak over de Pascal machine als de interpreter wordt bedoeld. De Pascal machine voert de door de compiler gegenereerde p-code uit. Om zo efficiënt mogelijk te werken is de interpreter vrij klein en zo snel mogelijk gemaakt. De instructies zijn dan ook voor de gebruiker niet erg duidelijk, maar dat hoeft ook niet, dit alles in tegenstelling tot een Basic interpreter.

Pascal compiler

De Pascal compiler is het vertaalprogramma dat de Pascal tekst, samengesteld met behulp van de editor, vertaalt naar de p-code. Omdat Pascal statements en p-code ver van elkaar afstaan is de compiler een groot en ingewikkeld programma, en daarom in Pascal geschreven. De gebruiker krijgt alleen de beschikking over de code van de compiler, helaas niet de source.

De UCSD Pascal compiler houdt zich over het algemeen behoorlijk aan de standaard Pascal zoals deze in „Pascal Manual and Report”, het boek dat Pascal definieert, staat beschreven. Er zijn een behoorlijk aantal ingrijpende uitbreidingen bij UCSD Pascal aanwezig.

Er zijn twee datatypen voorgedeclareerd, het type string (een packed array of characters met daaraan toegevoegd de lengte van de string) en het type long integer. De real heeft een nauwkeurigheid van 6 cijfers (4 byte floating point getal), wat voor sommige toepassingen, zoals administratieve, onvoldoende is. De long integer laat toe met integers met een in de praktijk onbeperkte grootte te werken, waardoor de gewenste precisie kan worden bereikt.

Andere uitbreidingen zijn de Unit en Segment mogelijkheden. Een Unit is een bij elkaar behorende groep procedures zonder een programma te vormen. Applestuff is zo'n Unit waarin typische Apple procedures zitten die vanuit elk Pascal programma zijn te gebruiken.

Segmentatie is een mogelijkheid om programma's die te groot voor het werkgeheugen zouden worden in segmenten op te delen, waarbij slechts een segment tegelijk in het werkgeheugen aanwezig hoeft te zijn, de rest blijft in het massageheugen: de floppy disk. De compiler zelf is een voorbeeld van een programma in segmenten. De compiler is te groot om in het werkgeheugen van de Apple te

Lijst

```
PROGRAM GOXY;
PROCEDURE FGOTOXY (X, Y: INTEGER);
(* HAZELTINE 1500 AND 1510 GOTOXY *)
VAR SEND: PACKED ARRAY [0..3] OF 0..255;
```

```
BEGIN
  IF X>79 THEN X:=79
  ELSE IF X<0 THEN X:=0;
  IF Y>23 THEN Y:=23
  ELSE IF Y<0 THEN Y:=0;
  SEND[0]:=126; (* LEAD-IN *)
  SEND [1]:=17; (* DC1 *)
  IF X<30 THEN SEND [2]:=X+96
  ELSE SEND [2]:=X;
  SEND [3]:=Y+96;
  UNITWRITE (2, SEND, 4);
END;
```

```
BEGIN (* DUMMY MAIN *)
END.
```

Lijst Voorbeeld van een GOTOXY-procedure.

passen. Dit is een praktische oplossing om kleine computers Pascal te laten verwerken.

Het GOTO-statement komt in UCSD Pascal in een beperkte vorm voor, alleen binnen in een procedure kan worden gesprongen. Vroegtijdig verlaten van een procedure is door EXIT mogelijk. De afkomst van UCSD Pascal, een leermiddel voor studenten, verraadt zich bij het GOTO-statement.

De compiler moet worden verteld dat GOTO-statements zijn toegestaan, standaard is GOTO niet toegestaan. Het is de bedoeling studenten te leren gestructureerd te programmeren en daarbij is GOTO uit den boze.

De in/uit verschillen tussen computers en de mogelijkheid van interactief programmeren bij microcomputers veroorzaken kleine verschillen tussen standaard Pascal en UCSD Pascal. In/uit-problemen zijn een bekend en vervelend, doch onvermijdelijk probleem in de computerwereld. In UCSD Pascal ontbreekt dispose, het vrijgeven van de ruimte die een dynamische variabele inneemt. In plaats daarvan zijn mark en release aanwezig, in andere Pascal implementaties komen we dat ook wel tegen.

Een goed geschreven standaard Pas-



UCSD Pascal

cal programma zal zonder veel wijzigingen door UCSD Pascal kunnen worden verwerkt. Uitwisselen tussen verschillende UCSD Pascal implementaties, en dat zijn er ondertussen al heel wat, is zonder meer mogelijk: UCSD Pascal is op elke machine in principe gelijk. De uitbreidingen van UCSD Pascal zullen het wel moeilijk maken UCSD Pascal programma's op andere Pascal implementaties te verwerken.

6502-assembler

Voor bijvoorbeeld tijdkritische problemen kan het nodig zijn een subroutine in de 6502-machinetaal te schrijven en deze subroutine vanuit een Pascal programma aan te roepen als externe procedure.

Voor dat doel is een assembler aan het operating system toegevoegd. Deze assembler heeft macro faciliteiten en kent conditional assembly mogelijkheden van het type IF... THEN...ELSE. Het is een uitstekende assembler.

Library en linker

In de library kunnen we een biblio-

theek van standaard programma's opbouwen. Vanuit een programma kunnen we deze programma's als externe programma's aanroepen. De linker zorgt ervoor dat het library programma met het gecompileerde programma wordt verbonden zonder dat het library programma opnieuw hoeft te worden gecompileerd.

De compiler activeert de linker automatisch als een extern programma wordt gedeclareerd.

Als we met de assembler een machinetaalsubroutine hebben geproduceerd, moet de gebruiker zelf de linker activeren om die routine aan het Pascal programma te hangen.

Documentatie

De mee geleverde documentatie is indrukwekkend.

Er zijn drie boeken over Pascal aanwezig in het documentatiepakket. Het standaard boek over Pascal „Pascal Manual and Report”, een onmisbaar werk.

UCSD Pascal, een dik boek met volledige informatie over het UCSD operating system op de Apple.

Een inleidend boek, geschreven voor UCSD Pascal door Kenneth Bowles van UCSD. Dit boek is niet voor iedere UCSD Pascal implementatie ge-

schikt, omdat in de programmavoorbeelden zogenoemde Turtle graphics worden toegepast. De grafische mogelijkheden van de Apple zijn echter voldoende om dit boek te gebruiken.

Conclusie

De implementatie van UCSD Pascal op de Apple is een welkome uitbreiding van de mogelijkheden van de Apple. Voor degenen die Basic een te beperkte taal vinden is Pascal en zeker de UCSD versie een goede taal. Dat UCSD Pascal op vele microcomputers wordt geïmplementeerd en een soort standaard aan het worden is, is ook een voordeel. Uniek is dat de Apple-versie met één disk drive kan werken, de meeste implementaties vereisen minimaal twee disk drives.

De beperkte eigenschappen van de terminal van de Apple zoals 40 karakters op een regel en alleen upper case zullen in de meeste gevallen geen ernstige beperking zijn, temeer daar via de paginatruc toch 80 karakters per regel mogelijk zijn. Een volwaardige terminal is ook aan te sluiten en zal in het gebruik wel beter bevallen.