NAME

pot − Format der Quelldateien für den **pot**(1) Assembler.

BESCHREIBUNG

Dieses Dokument erläutert das Format der Quelldateien, die der **pot**(1) Assembler verarbeiten kann.

ZEILENFORMAT

Eine **pot**(1) Quelldatei besteht aus Zeilen, von denen jede *Symbole*, *Befehle*, *Oktalzahlen* und *Kommentare* enthalten kann. Wenn eine Zeile zumindest einen *Befehl* oder eine *Oktalzahl* enthält, wird ein Speicherwert generiert und der Speicherzeiger erhöht.

Der Speicherzeiger verweist dabei auf die Speicherzelle, in die der nächste Wert abgelegt werden soll. Zu Beginn verweist der Speicherzeiger auf die Speicherzelle mit der Adresse null.

Der Speicherzeiger kann über zwei verschiedene Pseudobefehle manipuliert werden:

***** <*nnnn*>
: Setzt den Speicherzeiger auf den oktalen Wert <*nnnn*>.

PAGE
: Setzt den Speicherzeiger auf den Anfang der nächsten Speicherseite. Eine Speicherseite besteht aus 128 Worten.

SYMBOLE

Symbole stehen am Anfang einer Zeile. Damit ein Symbol korrekt erkannt wird, dürfen vor ihm höchstens Leer- und Tabulatorzeichen stehen. *Adressymbole* enden mit einem **:** (Doppelpunkt). Ihnen wird der aktuelle Wert des Speicherzeigers zugewiesen. *Wertsymbole* weden mit **=** abgeschlossen. Ihr Wert wird bestimmt, indem der Rest der Zeile ausgewertet und das Ergebnis dem Symbol zugewiesen wird. Der Speicherzeiger wird dabei nicht erhöht.

Wenn ein Symbol nicht durch **:** oder **=** terminiert wird, so gilt es als referenziert und wird duch seinen Wert ersetzt.

ANWEISUNGEN

Eine Anweisung besteht aus einem oder mehr *Befehlen*, *Symbolen* und/oder *Oktalzahlen*. Zusätzlich können Symbole und Oktalzahlen mit arithmetischen Operationen (+, -) zu Ausdrücken zusammengefasst werden. Der Ausdruck muß dazu in runde Klammern **( )** gesetzt werden. Ausdrücke werden berechnet und durch ihren Wert ersetzt. Die Befehle (Siehe **pep8**(7)) und Symbole werden durch ihre Werte ersetzt. Die sich ergebende Menge von Oktalzahlen wird oder-verküpft um ein einzelnes Anweisungswort zu erhalten.

Die Adressierung der Zero−Page wird automatisch erkannt und berücksichtigt. Sie bedarf keines speziellen Symbols. Indirekte Adressierung wird durch das spezielle Symbol **I** angezeigt.

TEXT

Mit dem Pseudobefehl **TEXT** kann Text als gepackte **sixbit(7)** Zeichen im Speicher abgelegt werden. Das erste Zeichen nach dem **TEXT** Befehl wird als Trennzeiche aufgefasst und darf im Text nicht vorkommen. Dieses Zeichen beendet den abzulegenden Text. Zunächst wird ein Wort mit der Anzahl der Zeichen im Text abgelegt, dann die Zeichen, je zwei pro Wort. Das erste Zeichen wird in den Bit 0-5 der ersten Wortes abgelegt, das zweite in Bit 6-11 der ersten Wortes. Dann wird das zweite Wort gefüllt usw.

Beispiel:

```
        TEXT    /HELLO WORLD!/      ; Begruessung
```

Beinhaltet zwölf Zeichen und erzeugt damit sieben Worte im Speicher.

KOMMENTARE

Kommentare werden durch **;** eingeleitet und erstrecken sich bis zum Ende der Eingabezeile.

BEISPIEL

```
;
; Aufaddieren der Werte einer Tabelle
;

CLLA=   CLA CLL                 ; Wertsymbol zuweisen
```

```
            JMP I       VEC     ; Erzeugt indirekten Sprung
VEC:        START               ; zum Programmstart

RESULT:     0                   ; Platz für das Ergebnis

TAB:        1                   ; Die
            3                   ; ersten
            5                   ; sechs
            7                   ; Primzahlen.
            13                  ; Natürlich
ENDT:       15                  ; oktal.

TLEN:       (ENDT-TAB+1)        ; Länge der Tabelle

TPTR:       TAB                 ; Zeiger zum Anfang
TEND:       ENDT                ; und Ende der Tabelle

            PAGE                ; Der Code beginnt auf
                                ; einer neuen Speicherseite

START:      RCL         RESULT  ; Zwischenergebiss holen
            CLL                 ; Link löschen
            TAD I       TPTR    ; Nächsten Wert addieren
            STO         RESULT  ; Neues Zwischenergebniss
            RCL         TPTR    ; Zeiger holen
            CLL CMA IAC         ; Zum Vergleichen subtrahieren
            TAD         TEND    ; Zeiger mit Endezeiger vergleichen
            SNA                 ; Nicht gleich? Weitermachen
            JMP         FINISH  ; Sonst fertig
            RCL         TPTR    ; Zeiger holen
            IAC                 ; erhöhen
            STO         TPTR    ; und zurückspeichern
            JMP         START   ; Nächster Wert
FINISH:     HLT
```

Nach dem Ausführen diese Programms steht in der Speicherzelle *0002* (Result) der Wert *0050*.

SIEHE AUCH

pot(1), **pepsi**(1), **pep8**(7)