NAME

pepsi − Der pep-8 Simulator

SYNOPSIS

pepsi [−vsti] [-o *addr*] [-d *addr:cnt* ...] [-e *dev,...*] [*datei*]

BESCHREIBUNG

Dieses Handbuch dokumentiert das **pepsi** Kommando. **pepsi** simuliert einen pep-8 Prozessor. Es kann ein **.pmi** Speicherabbild, welches durch den **pot**(1) Übersetzer erzeugt wurde geladenen und ausgeführt werden. Der Simulator wird beendet wenn er eine**HL T** Anweisung ausführt.

Wird kein Speicherabbild übergeben oder der Einzelschritt-Modus gewählt geht **pepsi** in den interaktiven Modus.

OPTIONEN

Optioen könne auf der Kommandozeile angegeben werden oder in einer Optionsdatei mit demselben Basisnamen wie das Speicherabbild aber der Endung **.pop** gelesen werden. Es wird nur die erste Zeile ausgewertet.

−v Ausführlicher (*verbose*) Modus. Vor der Ausführung einer Anweisung wird der Inhalt aller internen Register und ggf. der adressierten Speicherzelle ausgegeben. Ist der Quelltext-Modus aktiv, wird weiterhin die aktuelle Quelltextzeile ausgegeben.

−s Quelltext (*source*) Modus. Benutzt den Quelltext und die Symbole in der **.psy** Datei. Die ermöglicht die Benutzung von symbolischen Adressen. Vor dem Inhalt der Register wird die aktuelle Quelltextzeile ausgegeben.

Die Erzeugung der **.psy** Datei muß explizit angefordert werden (siehe **pot**(1)).

−t Einzelschritt (*trace*) Modus. Der Simulator hält vor jeder Anweisung an und geht in den interaktiven Modus. Der Einzelschritt Modus aktiviert automatisch den ausführlichen Modus. Der Quelltext-Modus muß ggf. zusätzlich aktiviert werden. Im Einzelschritt-Modus kann der Benutzer die angezeigte Anweisung ausführen, indem er die Eingabetaste (*return*) betätigt oder ein Kommando absetzen, um den Zustand des Simulators zu inspizieren.

−i Interaktiver (*interactive*) Modus. Enspricht dem Aufruf mit −st, also Quelltext- und Einzelschritt-Modus aktiviert.

−o addr

Angabe der Startadresse (*origin*) für die Simulation. Dabei kann *addr* eine Oktalzahl oder ein Symbol sein.

−d addr,cnt

Ausgabe (*dump*) von Speicherbereichen nach Ausführen einer **HLT** Anweisung. Es werden *cnt* Speicherworte beginnend mit *addr* in oktaler Notation ausgegeben. Dabei kann *addr* eine Oktalzahl oder ein Symbol sein. Die*−d* Option kann mehrf ach angegeben werden.

−e dev[*:params*],...

Aktivieren von einem oder mehreren Geräten. Für jedes Gerät wird die Gerätenummer und ggf. ein Parametersatz angegeben. Mehrere Geräte werden durch Komma getrennt.

−l Auflisten der unterstützten Geräte.

INTERAKTIVE KOMMANDOS

Im interaktiven Modus hält der Simulator an, bevor er eine Anweisung ausführt und gibt die aktuelle Quelltextzeile (wenn verfügbar) und den Zustand der internen Register in der Form *pc: addr: word -l:b ac: cont* aus. Dies sind:

Ausgabe	*Bedeutung*
pc: *addr*	Programmzähler (*program counter*)
word	Der Inhalt des aktuellen Speicherworts
l:*b*	Das Link-Bit
ac: *cont*	Der Akkumulator (*accumulator*)

Wenn die aktuelle Anweisung einen Operanden erfordert, wird dieser in der folgenden Zeile ausgegeben. Dies geschieht als Pfeil in der Form -(*XX*)-> wobei statt *XX* entweder **ZP** für die nullte Seite (*zeropage*) oder die Nummer der aktuellen Speicherseite bei Adressierung relativ zur aktuellen Seite

ausgegeben wird. Wenn es sich um eine indirekte Adresse handelt, wird ein zweiter Pfeil in der Form -(I)-> ausgegeben. Darauf folgt der Wert, der von der indirekten Adresse referenziert wird.

Wenn die aktuelle Anweisung ein Sprung, also **JMP** oder **JMS** ist, wird das Ziel des Sprungs ausgegeben. Wenn es ein indirekter Sprung ist, wird ein Pfeil in der Form -(I)-> ausgegeben, gefolgt vom Wert der referenziert wird.

Nach der Ausgabe der Register wird die aktuelle Beobachtungsliste abgearbeitet. Für jede beobachtete Adresse werden die entsprechenden Speicherworte ausgegeben. Dann wird eine Eingabeaufforderung gedruckt und es kann eines der folgenden Kommandos eingegeben werden:

<*return*>
: nächste Anweisung ausführen.

. *(Punkt)*
: Gibt den Zustand der internen Register und die aktuelle Quelltextzeile (erneut) aus.

b *addr* Unterbrechung (*breakpoint*) an der angegebenen Adresse setzen. Wird bei der Ausführung eine Unterbrechung erreicht, wird die Ausführung unterbrochen und in den interaktiven Modus geschaltet. Wird eine Unterbrechung auf den Beginn einer Unterroutine gesetzt, d.h. auf die Adresse in der die Rücksprungadresse gespeichert wird, wird die Ausführung nach dem Speichern der Rücksprungadresse und vor der Ausführung des ersten Befehls der Unterroutine unterbrochen. In diesen Fall werden sowohl die Quellzeile mit der Unterbrechung als auch die folgende Zeile (die die nächste ausgeführte Anweisung enthält) angezeigt.

bc *addr*
: Unterbrechung an der mit *addr* angegebenen Adresse löschen (*breakpoint clear*).

bl Unterbrechungen auflisten (*breakpoint list*).

c*base value*
: Einen Wert *value* mit Basis *base* umwandeln und ausgeben (*convert*). Gültige Angaben für *base* sind **o** (oktal), **d** (dezimal) und **h** (hexadezimal), der Wert wird jeweils in den anderen beiden Basen ausgegeben.

d *addr value*
: Den Wert *value* im Speicherwort an der Adresse *addr* speichern (*deposit*).

g Gehe bis zur nächsten Unterbrechung oder bis eine **HLT** Anweisung ausgeführt wird (*go*). Deaktiviert den ausführlichen und den Einzelschritt-Modus.

o *addr* Setze den Programmzähler auf die angegebene Adresse (*origin*).

p [*addr* **[***cnt***]]**
: Gibt die nächsten *cnt* Quelltextzeilen ab der angegebenen Adresse aus (*print*). Wird *cnt* nicht angegeben, werden zehn Zeilen ausgegeben. Wird auch die Adresse weggelassen, werden zehn Zeilen ab der aktuellen Adresse ausgegeben.

s Gibt die Symboltabelle aus (*symbols*).

w *addr* **[***cnt***]**
: Setzt die Adresse *addr* auf die Beobachtungsliste (*watch*). Wird die Beobachtungsliste angezeigt, werden *cnt* Speicherworte ausgegeben. Wird *cnt* weggelassen, wird nur ein Wort ausgegeben.

wc Löscht die Beobachtungsliste (*watchlist clear*).

x *addr* **[***cnt***]**
: Gibt *cnt* Speicherworte an Adresse *addr* aus (*examine*). Wird *cnt* weggelassen, wird nur ein Wort ausgegeben.

? Gibt einen kurzen Hilfetext aus.

q Beendet den Simulator (*quit*).

Adressen können entweder als absolute Adresse als Oktalzahlen mit bis zu vier Stellen oder in der Form *page:offset* mit separater Seitennummer und Offset innerhalb der Speicherseite angegeben werden. Seitennummer und Offset werden auch oktal angegeben. Im Quelltextmodus können statt numerischer Adressen auch die im Programm definierten Symbole verwendet werden.

EIN-/AUSGABE

Gerät 0 , Fernschreiber

Ist dieses Gerät selektiert verbindet sich der **pepsi**(1) mit einem simulierten Fernschreiber, der von **teletype**(1) realisiert wird.

Als Parameter bei der Aktivierung können Hostname und Port des **teletype**(1) Servers in der Form *host*:*port* angegeben werden. Werden die Parameter weggelassen wird *localhost*:*4200* benutzt. Wird **teletype**(1) ohne Parameter auf dem lokalen System gestartet wird ebenfalls Port 4200 benutzt, so dass im Regelfall ohne Parameter gearbeitet werden kann.

Gerät 1 , Lochstreifen-Leser

Mit dem Lochstreifen-Leser können Daten aus einer Datei des Host-Systems gelesen werden. Der Name dieser Datei wird als Parameter angegeben.

Gerät 2 , X/Y-Punkt Plotter

Dieses Gerät simuliert eine Osziloskopröhre mit einzeln ansteuerbaren Achsen. Über Einzelpulse können nachleuchtende Punkte an beliebigen Stellen gesetzt werden.

Das Gerät wird über die Java-App "Scope" realisiert mit der sich der Simulator verbindet.

Als Parameter bei der Aktivierung können Hostname und Port des **scope**(1) Servers in der Form *host*:*port* angegeben werden. Werden die Parameter weggelassen wird *localhost*:*4321* benutzt. Wird **scope**(1) ohne Parameter auf dem lokalen System gestartet wird ebenfalls Port 4321 benutzt, so dass im Regelfall ohne Parameter gearbeitet werden kann.

Gerät 3 , GPIO 7-Segment Anzeige

Auf Platformen mit GPIO Pins (Raspberry Pi) kann über diese Gerät einen an zwei GPIO Pins angschlossene 7-Segment Anzeige auf Basis eines TM1637 Controller-Chip angesteuert werden.

Als Parameter bei der Aktivierung kann der für "Clock" zu verwendene Pin angeben werden, der für "Data" verwendete ist automatisch der nächste. Wird der Parameter weggelassen werden die Pins 12 und 13 verwendet.

Gerät 4 , GPIO Ein- und Ausgabe

Auf Platformen mit GPIO Pins (Raspberry Pi) können über diese Gerät bis zu zwölf GPIO Pins als Ein- bzw. Ausgabe verwendet werden. Es werden die Pins 2 bis 14 angesteuert.

SIEHE AUCH

pot(1), **teletype(1), scope(1), pot**(5), **pep8**(7)

BUGS

Wahrscheinlich reichlich, aber noch unentdeckt.