

Ny version af SLAM-assembler

Pr. dags dato er en ny version af SLAM-assembleren på CYBER taget i brug.

De vigtigste ændringer/forbedringer er følgende:

- . index på variable/etiketter er udvidet, så der nu er mulighed for 256 variable pr. bogstav pr. blok mod tidligere 128
- . udskriften ved listoptions LIST, CONDLIST og TEXT er udvidet til også at omfatte hexadecimale adresser samt blokniveauet
- . som en option kan man få assembleren til at opfatte alt efter kolonne 72 eller kolonne 80 som kommentar. Formålet er at kunne få UPDATE-identifikatorer med på udskriften
- . udskriften ved brug af PRINT-option er ændret fra at udskrive det binære resultat af assembleringen til at skrive resultatet oktalt, decimalt og hexadecimalt
- . kommandoen S. er indført. Denne bevirker, at teksten efter S. bliver tolket som data, hvis iso værdier indsættes i den binære fil, i half-wordmode med en karakter/byte, i wordmode med en karakter/ord. Tal som indgår opfattes ikke som cifre men som tal.
Ønsker man flere konsekutive tal må de separeres med komma. Komma, som efterfølger tal, opfattes som blindt.
Bogstaver tolkes som små bogstaver.

En streng afsluttes med 3.

Eks.:

s. abcd3	=	97, 98, 99, 100, 3
s. 10, fejl3	=	10, 102, 101, 106, 108, 3
s. abc, 10, 3	=	97, 98, 99, 46, 10, 3

- . eftersom de beskrevne ændringer ville have medført et betydeligt merforbrug af ressourcer til assemblering, er der gennemført en række optimeringer, som tilsammen betyder, at en assemblering med options anvendt som hidtil vil få et noget mindre ressourceforbrug end tidligere, og at anvendelsen af options, som giver UPDATE-identifikatorer på udskriften, vil give et ressourceforbrug, som er sammenligneligt med tidligere.

Eksempler i forbindelse med UPDATE.

UPDATE, P, Q, D, L = ...

COMPILE filen indeholder linier med 80 tegns tekst efterfulgt af 10 tegns komprimeret UPDATE linieidentifikator. Assembleres med

PASCAL, LOAD=SLAM,...BINARY,CONDLST, 80.

som indsætter et semikolon i kolonne 80 i alle linier, som er længere end 79 tegn.

. UPDATE, P, Q, L = ...

COMPILE filen indeholder linier med 72 tegns tekst efterfulgt af 18 tegns UPDATE linieidentifikator.

Assembleres med

PASCAL, LOAD=SLAM,...BINARY,CONDLST, 72.

som indsætter et semikolon i kolonne 72 i alle linier, som er længere end 71 tegn.

Keld Schmidt-Møller

SLAM-programmeringskonventioner til RC 3500 med monitor 1

Navnekonventioner

- 1) a, b, c og d-navnene er lokale for et enkelt modul (korutine eller driver) og kan anvendes efter eget forgodtbefindende. Eksempelvis kan nævnes følgende anvendelser:

a - lokale labels i en blok inden for et enkelt modul.

b - lokalt globale labels inden for et enkelt modul.

c, d - konstanter, parametre, pointere eller offsets i datastrukturer o.s.v., som eventuelt assignes via e-navnene (se næste punkt).

- 2) e-navnene anvendes inden for et enkelt modul som:

I: pseudonymer for størrelser, som kan (eller skal) kommunikeres til og fra andre moduler.

II: konstanter, parametre, pointere eller offsets i datastrukturer o.s.v. (se eventuelt punkt 1) for c og d-navnene).

- 3) f og g-navnene anvendes globalt til at kommunikere størrelser mellem forskellige moduler på compile time.

- 4) h, i og j-navnene er reserveret monitoren. Ubenyttede navne kan i nødsfald anvendes som globale navne, hvis man løber tør for f og g-navne.

Operationer

Der skelnes mellem to typer operationer:

- 1 - simple operationer,
- 2 - buffer operationer.

Simple operationer har formatet:

P + J10 chain (kræves af monitoren)
 + J11 answer semaphore (kræves, hvis operationen sættes i
 monitorkøen)

Buffer operationer har formatet:

P + J10 chain (kræves af monitoren)
 + J11 answer semaphore (kræves, hvis operationen sættes i
 monitorkøen)
 + J12 top transferred
 + J14 first of buffer
 + J16 top of buffer
 + J13 status
 + J15 code

Ovenstående formater er standard, og "private" udvidelser sker "nedad".
 Udvidelser "opad" er reserveret eventuelle monitorudvidelser.

Korutiner

En korutine består af et dataområde og et stykke kode, som kan være fælles for flere korutine-inkarnationer.

Koden skal bestå af en blok, som er åben med hensyn til e-navne. Reset-koden kan placeres sidst i denne blok. Dataområdet placeres uden for kode-blokken i en omgivende e-blok.

Den omgivende e-blok består altså af:

- 1) e-pseudonym definitioner
- 2) dataområder hørende til eventuelt flere inkarnationer af
 korutinen

Reset-koden kan eksempelvis laves på følgende måder:

- 1) Reset af flere inkarnationer foregår i en løkke, idet data-områderne kædes sammen med et kædeord i korutine-beskrivelsen.
- 2) Reset af flere inkarnationer foregår ved at resette en inkarnation og hoppe ud i dataområdet, hvorfra man enten hopper tilbage til reset eller til reset af næste korutine.

Eksempel

b. f127, g127

<monitor>

b. e17

ldr w0 i

e0; goto reset

; e-names

e3 = f17

coroutine
description 1

coroutine
description n

b. b23

e0 = b0

b. a17, c20, d18

e

b0:

; reset:

e.

f2 = e10

e.

```

b. e8
    ldr w0 i
    e3 = f2
    fl7 = e7
    coroutine description
    b. b8
    e0 = b0
    e.
e.
h12:

```

Gode vaner

- korutinerne kodes reentrant af effektivitetsgrunde.
- siderne inden for et modul nummereres.
- hver side kode afsluttes med STOPCODE.
- en kode-linie har formatet:

(6) 5	3	3	2	1	2	1	2	1	n	1	4	1	m
				SP		SP							
LABEL	OP		wi	IN	xi	+	-	EXPRESSION	SP	LABEL	;	COMMENTS	
		a					SP		,				
		b											
		c											
		e											
		d											
		s.											

Pascal

512
Printer

731
printer

CRT

53

+	+	+	+
-	-	-	-
*	*	*	*
/	/	/	/
(	(	(	(
)	)	)	)
\$	\$	\$	\$
=	=	=	=

,	,	,	,
---	---	---	---

'	≡	≠	#	#
---	---	---	---	---

[	Æ	[	[
---	---	---	---

]	Ä	]	]
---	---	---	---

:	:	:	:
---	---	---	---

≠	≡	"	"
---	---	---	---

{	→	7	-	-
---	---	---	---	---

√	≤	!	!
---	---	---	---

∧	]	&	&
---	---	---	---

↑	┌		
---	---	--	--

}	↓	≥	?	?
---	---	---	---	---

<	<	<	<
---	---	---	---

>	>	>	>
---	---	---	---

≤	┌	@	@
---	---	---	---

≥	∅	\	\
---	---	---	---

┐	∧	∧	∧
---	---	---	---

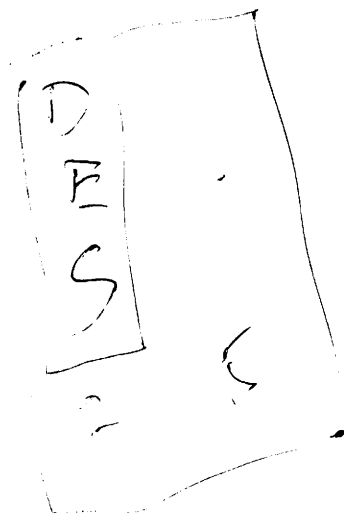
j	j	j	j
---	---	---	---

77


```

10 finis          ksm0 at 10 30
20      1      0 000000 0000  b. n10
30      2      0 000000 0000  ist w1 s x1  0  b-format
40      3      2 000002 0002  ist w1 s x1 31  c
50      4      6 000006 0006  ist w1 s x0 31  c
60      5     10 000012 000a  ist w1      x1 31  a
70      6     12 000014 000c  ist w1      x0 31  c
80      7     16 000020 0010  ist w1 s x1 63  c
90      8     20 000024 0014  ist w1 s x0 63  c
100     9     24 000030 0018  ist w1      x1 63  a
110    10     26 000032 001a  ist w1      x0 63  c
120    11     30 000036 001e  e.
130

```



Same v.

index undefined

Series v. udskrift

TERMINAL VJ

1010 2 0 000000 0000 171373 f2fb
1020 3 2 000002 0002 175373 fafb
1030 3 4 000004 0004 000037 001f
1040 4 6 000006 0006 175173 fa7b
1050 4 8 000010 0008 000037 001f
1060 5 10 000012 000a 155237 da9f
1070 6 12 000014 000c 175033 fa1b
1080 6 14 000016 000e 000037 001f
1090 7 16 000020 0010 175373 fafb
1100 7 18 000022 0012 000077 003f
1110 8 20 000024 0014 175173 fa7b
1120 8 22 000026 0016 000077 003f
1130 9 24 000030 0018 155277 dabf
1140 10 26 000032 001a 175033 fa1b
1150 10 28 000034 001c 000077 003f
1160 11 30 000036 001e 777777 ffff

1170

1180

1190 11 30 000036 001e prog end; a: 2; b: 1; c: 6

1200 errorcount: 0; swapcount: 1