



DECUS

PROGRAM LIBRARY

DECUS NO.	FOCAL8-59
TITLE	FOCAL OVERLAY COMMON AREA FOR 4K CORE MEMORY
AUTHOR	Herbert Zimmermann
COMPANY	Digital Equipment GmbH Köln, Germany
DATE	Submitted December 8, 1969
SOURCE LANGUAGE	FOCAL

FOCAL OVERLAY COMMON AREA FOR 4K CORE MEMORY

DECUS Program Library Write-up

DECUS No. FOCAL8-59

ABSTRACT

The common area is implemented with the function FNEW in the following way:

SET Z = FNEW(U,V,W)

U = 1 means store into common area

U = 2 means fetch from common area

V = 1 to 30 is the calling number of the data (storage area has a maximum of 30 datas)

W is the data in the floating point format

Two versions of the overlay tape are available:

1. For FOCAL without extended functions. The used area in core memory is 5071_8 to 5357_8 .
2. For FOCAL with extended functions. The used area in core memory is 4271_8 to 4557_8 .

The overlay "common area" is read in in a normal way:

Read in of FOCAL

Do initial dialogue

Stop computer

Read in of the overlay (SA = 7777)

Start at SA = 200

THE IMPLEMENTATION OF A "COMMON AREA" IN FOCAL WITH THE FUNCTION FNEW

The main function of a "common area" is to safe a set of data in a way that they cannot be erased with a normal erase command.

Normally a "common area" is a prerequisite for chaining of user programs.

MINIMUM HARDWARE CONFIGURATION

PDP-8/I or PDP-8/L (or computers of the PDP-8 family) with 4K memory.

SOFTWARE NEEDED

Binary loader

FOCAL version: DEC-08-AJAE-PB

DESCRIPTION

FOCAL consists of the interpreter program and the initial dialogue. Via the initial dialogue the user tells the interpreter program whether he needs the extended functions or not! If the user types "No" after all questions which are asked by FOCAL, he will have a user area from 3220_8 to 5377_8 .

The content of the location 0035_8 , called Bottom, is used by FOCAL to determine the last free address of the user area. Therefore the content of Bottom must be 5377_8 .

If the user types "Yes" after all questions which are asked by FOCAL he will have a user area from 3220_8 to 4617_8 . The content of the location Bottom will then be 4617_8 .

The additional function FNEW will be stored in the user area. Therefore Bottom must be corrected.

DEFINITION OF THE FUNCTION FNEW

With FNEW the user must be able to store and fetch a maximum of 30 datas with floating point format in the "common area." The user must have random access to each data.

Therefore $FNEW(U, V, W)$

U = 1 means store

U = 2 means fetch

V = 1 to 30 means the calling number of the data

W is the data in floating point format.

Now we can write the following small program which asks for datas and stores them in the "common area."

```
1.01 ERASE
1.03 FOR V = 1, 30; DO 2
1.05 QUIT
2.05 ASK A(V)
2.10 SET U = 1; SET W = A(V)
2.15 SET Z = FNEW(U, V, W)
      Z is a dummy variable
```

To get the datas out of the common area we may write the following small program.

```
1.01 ERASE
1.03 FOR V = 1, 30; DO 2
1.05 QUIT
2.05 SET U = 2
2.10 SET W = FNEW (U, V, Z)
2.15 TYPE %, W, !
```

SHORT DESCRIPTION OF THE PROGRAM FNEW
(FOCAL without extended functions)

The common area is within the range of 5230_8 to 5377_8

1) Patch the address Bottom
The new value is 5070_8

Therefore *0035
 BOTTOM, 5070

2) FOCAL must know where the start address of the new program is. The subroutine FNEW is called XFNEW. When FOCAL has recognized FNEW in a user program it looks at the address 0410_8 . The content of this address must be identical with the start address of the subroutine XFNEW.

Therefore *0410
 XFNEW
 *5230
XFNEW, XXXX

3) After FOCAL has recognized FNEW and found it valid it evaluates the first argument in the parenthesis and provides the value of this argument for further action.

Question: Where can we find the value of this argument?

Answer: The value of the argument U is stored in the floating accumulator of the floating point package. The floating accumulator has the following addresses:

$0044_8 = \text{FLAC}$
 $0045_8 = \text{FLAC} + 1$
 $0046_8 = \text{FLAC} + 2$

Question: Which format has the data in the "floating AC?"

Answer: FLAC contains the exponent
 FLAC + 1 contains the high order part of the mantissa
 FLAC + 2 contains the low order part of the mantissa

Question: We don't need the floating point format. We need the integer value of the "floating AC." How can we get the integer value?

Answer: After FOCAL has recognized FNEW, the program jumps indirect via address 0410_8 to XFNEW. The integer value of the "floating AC" is transferred to the accumulator with the command:

JMS I Integer

therefore

XFNEW, CLA
 JMS I Integer

Question: Where can we find the subroutine Integer?

Answer: The address 0053_8 contains the start address of the subroutine Integer

therefore $INTEGER = 0053$

This means if $U = 1$ the accumulator contains value 0001_8

4) After the Evaluation of U we have to evaluate the argument V

Question: How can we store the value of the argument V in the "floating AC?"

Answer: First we have to jump over a space which may be in the parentheses and a comma. This will be done with

SPNOR
GETC

Both commands are subroutines which can be reached with indirect JMS. The addresses which contain the start addresses of the subroutine are located in page \emptyset . Therefore

SPNOR = JMS I XSPNOR
= JMS I 160
= 4560_8

GETC = JMS I UTRA
= JMS I 145
= 4545

Now we evaluate the argument V with the commands

PUSHJ
EVAL

PUSHJ is a jump to the subroutine XPUSHJ. EVAL is an address which is used within the subroutine XPUSHJ. Therefore

PUSHJ = JMS I XPUSHJ
= JMS I 140
= 5540
EVAL = 1613

After the execution of PUSHJ; EVAL the value of the argument V is in the "floating AC. "

5) After we have used the value of the argument V in our subroutine XFNEW we need the value of the argument W.

Question: How can we store the value of the argument W in the "floating AC?"

Answer: Similar to 4)

```
SPNOR
GETC
PUSHJ
EVAL
```

6) To get a data from the "common area" we must do the following. We store the data from the common area in the "floating AC."

```
TAD Data 1 / Exponent
DCA FLAC
TAD Data 2 / High order mantissa
DCA FLAC + 1
TAD Data 2 / Low order mantissa
DCA FLAC + 2
JMP I EFUN
```

The return to the FOCAL user program is done by JMP I EFUN

```
JMP I EFUN
EFUN = 01638
```

The command JMP I EFUN is doing the following:

The "floating AC" will be combined with the variable name in the FNEW function and will be stored somewhere in the user area

```
SET Z = FNEW( )
after return
  (FLAC
Z = (FLAC + 1
  (FLAC + 2
```

7)

Question: How can we type an error message on the teletype during execution of XFNEW?

Answer: We store the ASCII sign, which we will type, in the accumulator and give the command JMS I Outdev. After execution of JMS I Outdev return goes to the address after the instruction.

```
JMS I OUTDEV
OUTDEV = 00638
```

Question: How can we stop the execution of XFNEW?

Answer: With

```
JMP I FOST
FOST, 02008
```

FOST is the start address 200₈ of FOCAL

8)

Question: How can we modify the program FNEW to use it with FOCAL with the extended functions?

Answer: Modify:
Bottom
XFNEW
AM 1
AM 2
STTABL

9)

Question: If we need a smaller or greater "common area," what can we do?

Answer: Modify:
Bottom
MMaxsp / negative, maximum data number
STTABL / start address of the common area.

10)

Question: How can we read the program "common area" into core memory?

Answer: Use RIM loader
Bin loader

FOCAL
Do initial dialogue
Stop the computer
Read in the paper tape binary FOCAL overlay "common area"
Start FOCAL manually with SA = 200

11)

Question: Where can we get additional information regarding to FOCAL?

Answer: Buy
Manual: Advanced FOCAL, technical specifications
DEC-08-AJBB-DL
FOCAL listing: DEC-08-AJAE-LA
Ask local software support specialist
If not enough, wait:
FOCAL is open ended, new informations will currently be issued.

LISTING
 FOCAL OVERLAY "COMMON AREA"
 FOR FOCAL WITH EXTENDED FUNCTIONS (4K)
 PATCH ADDRESS 4476 FROM 7041 TO 7040

AM1	4550
AM2	4551
BOTTOM	0035
CNTR	4555
EFUN31	0136
END	4452
EVAL	1613
FLAC	0044
FOST	0200
FRT1	4457
FRT2	4502
GETC	4545
INTEGE	0053
MMAXSP	4554
M1	4544
M2	4545
OUTDEV	0063
OUT01	4453
OUT02	4455
PNTR1	4552
PNTR2	4557
PUSHJ	4540
PI	4546
QMARK	4547
QPRINT	4445
SPNOR	4560
STTABL	4556
UBUF	4543
VBUF	4553
XAM1	4517
XAM2	4531
XFNEW	4430

0035	4270	BOTTOM,	*0035 4270
0410	4430		*410 XFNEW
4430	7200	XFNEW,	*4430 CLA
4431	4453		JMS I INTEGER
4432	3343		DCA UBUF
4433	1343		TAD UBUF
4434	1344		TAD M1
4435	7650		SNA CLA
4436	5253		JMP OUT01

4437	1343		TAD UBUF
4440	1345		TAD M2
4441	7650		SNA CLA
4442	5255		JMP OUT02
4443	1346		TAD PI
4444	4463		JMS I OUTDEV
4445	7200	QPRINT,	CLA
4446	1347		TAD QMARK
4447	4463		JMS I OUTDEV
4450	7200		CLA
4451	5652		JMP I END
4452	0200	END,	FOST
4453	1350	OUT01,	TAD AM1
4454	5257		JMP FRT1
4455	1351	OUT02,	TAD AM2
4456	5257		JMP FRT1
4457	3352	FRT1,	DCA PNTRI
4460	4560		SPNOR
4461	4545		GETC
4462	4540		PUSHJ
4463	1613		EVAL
4464	4453		JMS I INTEGER
4465	7450		SNA
4466	5245		JMP QPRINT
4467	3353		DCA VBUF
4470	1353		TAD VBUF
4471	1354		TAD MMAXSP
4472	7700		SMA CLA
4473	5245		JMP QPRINT
4474	1353		TAD VBUF
4475	1344		TAD MI
4476	7041		CIA
4477	3355		DCA CNTR
4500	7000		NOP
4501	3353		DCA VBUF
4502	2353	FRT2,	ISZ VBUF
4503	2353		ISZ VBUF
4504	2353		ISZ VBUF
4505	2355		ISZ CNTR
4506	5302		JMP FRT2
4507	1353		TAD VBUF
4510	1356		TAD STTABL
4511	3357		DCA PNTR2
4512	4560		SPNOR
4513	4545		GETC
4514	4540		PUSHJ
4515	1613		EVAL
4516	5752		JMP I PNTRI
4517	7200	XAM1,	CLA

4520	1044		TAD FLAC
4521	3757		DCA I PNTR2
4522	2357		ISZ PNTR2
4523	1045		TAD FLAC +1
4524	3757		DCA I PNTR2
4525	2357		ISZ PNTR2
4526	1046		TAD FLAC +2
4527	3757		DCA I PNTR2
4530	5536		JMP I EFUN
4531	7200	XAM2,	CLA
4532	1757		TAD I PNTR2
4533	3044		DCA FLAC
4534	2357		ISZ PNTR2
4535	1757		TAD I PNTR2
4536	3045		DCA FLAC +1
4537	2357		ISZ PNTR2
4540	1757		TAD I PNTR2
4541	3046		DCA FLAC +2
4542	5536		JMP I EFUN.
		INTEGER=0053	
4543	0000	UBUF,	0
4544	7777	M1,	-1
4545	7776	M2,	-2
4546	0261	P1,	0261
		OUTDEV=0063	
		FOST=0200	
4547	0277	QMARK,	0277
4550	4517	AM1,	4517
4551	4531	AM2,	4531
4552	0000	PNTR1,	0
		SPNOR=4560	
		GETC=4545	
		PUSHJ=4540	
		EVAL=1613	
4553	0000	VBUF,	0
4554	7741	MMAXSP,	7741
4555	0000	CNTR,	0
4556	4271	STTABL,	4271
4557	0000	PNTR2,	0
		FLAC=0044	
		EFUN =0136	

LISTING
 FOCAL OVERLAY "COMMON AREA"
 FOR FOCAL WITHOUT EXTENDED FUNCTIONS (4K)
 PATCH ADDRESS 5276 FROM 7041 TO 7040

AM1	5350
AM2	5351
BOTTOM	0035
CNTR	5355
EFUN	0136
END	5252
EVAL	1613
FLAC	0044
FOST	0200
FRT1	5257
FRT2	5302
GETC	4545
INTEGE	0053
MMAXSP	5354
M1	5344
M2	5345
OUTDEV	0063
OUT01	5253
OUT02	5255
PNTR1	5352
PNTR2	5357
PUSHJ	4540
PI	5346
QMARK	5347
QPRINT	5245
SPNOR	4560
STTABL	5356
UBUF	5343
VBUF	5353
XAM1	5317
XAM2	5331
XFNEW	5230

0035	5070	BOTTOM,	*0035 5070
0410	5230		*410 XFNEW
5230	7200	XFNEW,	*5230 CLA
5231	4453		JMS I INTEGER
5232	3343		DCA UBUF
5233	1343		TAD UBUF

5234	1344		TAD M1
5235	7650		SNA CLA
5236	5253		JMP OUT01
5237	1343		TAD UBUF
5240	1345		TAD M2
5241	7650		SNA CLA
5242	5255		JMP OUT02
5243	1346		TAD P1
5244	4463		JMS I OUTDEV
5245	7200	QPRINT,	CLA
5246	1347		TAD QMARK
5247	4463		JMS I OUTDEV
5250	7200		CLA
5251	5652		JMP I END
5252	0200	END,	FOST
5253	1350	OUT01,	TAD AM1
5254	5257		JMP FRT1
5255	1351	OUT02,	TAD AM2
5256	5257		JMP FRT1
5257	3352	FRT1,	DCA PNTR1
5260	4560		SPNOR
5261	4545		GETC
5262	4540		PUSHJ
5263	1613		EVAL
5264	4453		JMS I INTEGER
5265	7450		SNA
5266	5245		JMP QPRINT
5267	3353		DCA VBUF
5270	1353		TAD VBUF
5271	1354		TAD MMAXSP
5272	7700		SMA CLA
5273	5245		JMP QPRINT
5274	1353		TAD VBUF
5275	1344		TAD M1
5276	7041		CIA
5277	3355		DCA CNTR
5300	7000		NOP
5301	3353		DCA VBUF
5302	2353	FRT2,	ISZ VBUF
5303	2353		ISZ VBUF
5304	2353		ISZ VBUF
5305	2355		ISZ CNTR
5306	5302		JMP FRT2
5307	1353		TAD VBUF
5310	1356		TAD STTABL
5311	3357		DCA PNTR2

5312	4560		SPNOR
5313	4545		GETC
5314	4540		PUSHJ
5315	1613		EVAL
5316	5752		JMP I PNTR1
5317	7200	XAM1,	CLA
5320	1044		TAD FLAC
5321	3757		DCA I PNTR2
5322	2357		ISZ PNTR2
5323	1045		TAD FLAC +1
5324	3757		DCA I PNTR2
5325	2357		ISZ PNTR2
5326	1046		TAD FLAC +2
5327	3757		DCA I PNTR2
5330	5536		JMP I EFUN
5331	7200	XAM2,	CLA
5332	1757		TAD I PNTR2
5333	3044		DCA FLAC
5334	2357		ISZ PNTR2
5335	1757		TAD I PNTR2
5336	3045		DCA FLAC +1
5337	2357		ISZ PNTR2
5340	1757		TAD I PNTR2
5341	3046		DCA FLAC +2
5342	5536		JMP I EFUN
		INTEGER=0053	
5343	0000	UBUF,	0
5344	7777	M1,	-1
5345	7776	M2,	-2
5346	0261	P1,	0261
		OUTDEV=0063	
		FOST=0200	
5347	0277	QMARK,	0277
5350	5317	AM1,	5317
5351	5331	AM2,	5331
5352	0000	PNTR1,	0
		SPNOR=4560	
		GETC=4545	
		PUSHJ=4540	
		EVAL=1613	
5353	0000	VBUF,	0
5354	7741	MMAXSP,	7741
5355	0000	CNTR,	0
5356	5071	STTABL,	5071
5357	0000	PNTR2,	0
		FLAC=0044	
		EFUN =0136	