ASS ASSEMBLER

Version 1.0 (2 May 2021)

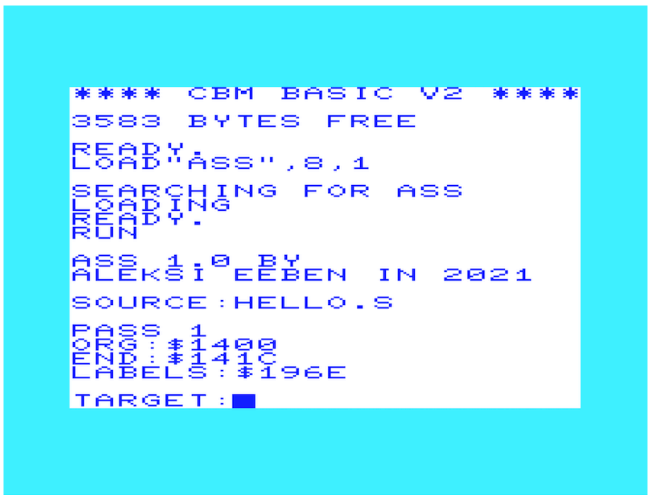
Symbolic 2-Pass Assembler for VIC 20, C-64, C-16/Plus4, C-128 and PET

by Aleksi Eeben in 2021 / aleksi.eeben@gmail.com

Quickstart

```
1000 :*************
1005 : HELLO WORLD
1010 :*************
1020 ORG $1800
1030 CHROUT=$FFD2
1040 LDY #$00
1050 LOOP:
1060 LDA TEXT,Y
1070 BEQ END
1080 JSR CHROUT
1090 INY
1100 JMP LOOP
1110 END:
1120 RTS
1130 TEXT:
1140 DB "HELLO WORLD",$0D,$00
```

```
SAVE"HELLO.S",8,1

LOAD"ASS",8,1

RUN

ASS 0.9
SOURCE:HELLO.S
.
.
TARGET:HELLO
.
.

LOAD"HELLO",8,1

SYS6144
```



```
**** CBM BASIC V2 ****
3583 BYTES FREE
READY.
LOAD"ASS",8,1
SEARCHING FOR ASS
LOADING
READY.
RUN
ASS 1.0 BY
ALEKSI EEBEN IN 2021
SOURCE:HELLO.S
PASS 1
ORG:$1400
END:$141C
LABELS:$196E
TARGET:█
```

Ass Style: Comments, Labels, Prefixes

Comments

```
1000 : Use double stop for full line comments or to insert empty lines
1010 :
1020 : Unexpanded VIC 20 Lite version accepts full line comments only
1030 :
1040 LDA #$00    ; In other versions you can add
1050 STA $D020   ; comments after a semicolon
```

Labels

Maximum label length is 10 characters, but tokenized basic words count as 1 character each (DATA, PRINT, LIST, RUN, INPUT, AND, OR, END, etc.)

```
1000 : Branch labels must end to a double stop. Label must end the line
1005 :
1010 LOOP:
1020 JMP LOOP
```

```
1000 : Define a label using '=' (no spaces)
1005 :
1010 CHROUT=$FFD2
```

8-bit offset can be added to or subtracted from a label at compile time:

```
1010 LDA LABEL+$F0,X
1020 LDA LABEL-$D8,Y
```

Simple label arithmetic is possible, but redefining is not allowed:

```
1020 LINE1=LABEL+$CE  ; OK
1030 LINE2=LINE1+$28  ; OK, but LINE1=LINE1.. would cause DUPLICATE LABEL
```

Use '<' or '>' prefix to take low byte or high byte of a label value:

```
1000 LDA #<IRQCODE    ; Take low byte of IRQCODE address
1010 STA $0314
1020 LDA #>IRQCODE    ; Take high byte of IRQCODE address
1030 STA $0315
```

Use '!' prefix to explicitly expand a 8-bit label to 16 bit addressing:

```
1000 LDY !BYTE,X      ; Compiles to LDY $00nn,X instead of LDY $nn,X
1010 LDY <BYTE,X      ; Similarly, this always becomes a 8-bit LDY
```

Ass Style: Numbers, Strings, Directives

Hex, Binary

$1C	8-bit values: $ + 2 hex digits
$D400	16-bit values: $ + 4 hex digits
%10110010	Byte in binary

Decimal mode is not currently supported

DB, DW

```
1000 : Bytes
1010 DB "ASCII TEXT",$00
1020 DB "MORE ",$10,$80,<LABEL,>LABEL,$00," MORE TEXT",$00

1000 : Words
1010 DW LABEL,$1234,$ABBA,$0000,$FFFF
```

LIB

Include another source file or a LIBrary

```
1000 ORG $1800
1010 LIB "KERNAL"
```

Alas, LIB calls cannot be nested due to 1541 limitation of max. 3 open files simultaneously. This needs a rewrite, maybe adding a preprocessor?

BIN

Embed a BINary file at the current location

```
1000 ORG $1FFE
1010 BIN "FONT"
```

ORG

Set target memory address. This should be on the first line of your code

```
1000 ORG $1800
```

Multiple ORG's add zero padding to the target file until the new address is reached. ORG areas must be in order from low to high memory and cannot overlap

Ass Errors

Full Version

ORG BACKWARD	New ORG overlaps with the previous ORG. ORG's must be in order from low to high memory
FILE ERROR	File not found or any other file error
ERROR	Syntax error
BRANCH TOO FAR	BNE, BEQ, etc. reaching too far, beyond the 8-bit range
DUPLICATE LABEL	A label with the same name was defined again. New definition is ignored
UNDEFINED LABEL	A non-existing label was referenced
PHASE ERROR	Instruction sizes in pass 1 and pass 2 don't match. This is often caused by forward references in zeropage. Use ! or < prefix to define labels as 16-bit or 8-bit during pass 1 or define zeropage addresses before using them

Unexpanded VIC 20 LITE Version

BRA	BNE, BEQ, etc. reaching too far, beyond the 8-bit range
DUP	Duplicate label
ERR	Any other error
(PHASE ERROR)	The end address is shown after the last line number after each pass. If the end addresses don't match, there is a phase error (or possibly a disk error)

Ass Builds

Full Version

ASS20	Unexpanded VIC 20
ASS20-3K	VIC 20 with 3K memory expansion
ASS20-8K	VIC 20 with memory expansion of 8K or more
ASS64	Commodore 64

Special Builds

ASS20LITE	Minimal Unexpanded VIC 20 version **
ASS20-40960	VIC 20 with RAM in Block 5, sys40960 to run ***
ASS64-49152	Commodore 64 high memory, sys49152 to run ***

Other Commodore Machines

ASSPET	Commodore PET *
ASS16PLUS4	Commodore 16/Plus4 *
ASS128	Commodore 128 *

* Automated, non-tested builds

** LITE version is under 2K in size and contains only minimal error messages and error checking. This leaves 300+ bytes more memory for user labels on an unexpanded VIC 20

*** High memory versions have much less space for user labels than regular basic area versions, because labels are located right after the compiler code

C'mon machine code monitor (VIC 20 and C-64) I wrote 20 years ago is included on the test disk for your convenience.

Happy assembling!