ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.
ALBIN, A PDP-8 LOADER FOR RELOCATABLE BINARY PROGRAMS

DECUS Program Writeup

Abstract

As part of the current design of a PDP-8 real-time monitoring system, a simple method has been obtained to construct relocatable binary formatted programs, using the PAL III Assembler. Allocation of these programs can be varied in units of one memory page (128 registers). While loading an ALBIN program the actual absolute addresses of indicated program elements (e.g. the keypoints of subroutines) are noted down in fixed program-specified locations on page 0. In order to make a DEC symbolic program suitable for translation into its relocatable binary equivalent, minor changes are required; which, however, do not influence the length of the program. Due to its similarity to the standard DEC Binary Loader, the ALBIN loader is also able to read-in normal DEC binary tapes. The loader is presented here in its simplest form, although the loading method, in a slightly advanced manner, will be used for automatic "piling while loading" of arbitrary sequences of more or less independent programs. In the form to be described, ALBIN requires 122\textsubscript{10} locations, including the RIM loader. Piling up in core memory of ALBIN programs stored on conventional or DECTape can be achieved using the same method with minor modifications.

Introduction

When a selected set of programs have to be executed in one run, this normally means full assembling of the set stacked together in the desired combination although a binary equivalent of each program is present. This inconvenience becomes a serious limitation in our application, the design of a parallel multiprocessor system. We, therefore, constructed a loading method which enables the loading of an arbitrary sequence of binary program tapes.

This implies that a binary program tape has to be loaded from a relocatable starting address. The problem with relocation is that the binary form of some program elements depends on the starting address of the program. If we, however, restrict ourselves to relocations in units of one memory page, and if we assume that the page 0 parts of relocatable programs are not to be relocated, the only changeable binary program elements are absolute (12 bits) addresses, referring to a relocatable program point.

Description of ALBIN - Format

For a binary loader, it is rather difficult to differentiate between 12-bit addresses and instructions unless the loader receives additional information from the assembler.

A DEC binary format tape consists of origin settings and memory words. A relocatable binary tape should not contain any origin setting, since this information changes with every loading. Instead, one may insert the origin information via the switch
register. In doing so, the origin setting symbol can be used for "key setting," which then implies that "\textasteriskcentered ADRES" means: set the address of the next instruction in ADRES.

In this way we can handle the absolute (12 bit) addresses. If the register REG should contain an absolute address referring to a relocatable program point AA, we place just above AA the expression "\textasteriskcentered REG". While loading, the ALBIN loader then places the right value of AA in REG, for example:

\[
\begin{align*}
\text{REG, AA} & \quad \text{\textasteriskcentered REG} \\
\text{AA, } & \text{\textasteriskcentered } \\
\end{align*}
\]

The condition must be fulfilled that REG is an address in the same relocatable program as \textasteriskcentered REG, or that REG is fixed (c.q. page 0) address. Therefore, it is impossible to have references between independently relocatable programs over a key address which belongs to one of them. References between independently relocatable programs, therefore, should occur over fixed key addresses at page 0.

Applications

One of the essential advantages of a relocatable loading system is the possibility of automatic piling of programs. In that case the above mentioned switch register setting is replaced by a register in the loader itself which, after each loading, points to the next free page.

Applying a "chaining" principle, one can also load and "remove" less frequently used routines while running (segmenting). This is particularly useful if magnetic tape input is available.

This work is preparatory for a software project performed under the direction of R. van Dantzig.

Directions of Use

1. Add to the permanent symbol table of PAL III the symbols:

   INPUT = 7545  
   ALBIN = 0

2. The program to be loaded must exist of one or more blocks separated by leader trailer.

   A block can be either a BIN or an ALBIN block. At the beginning of each block, the loader is automatically set into BIN mode. It can be brought into ALBIN mode by:

   \textasteriskcentered INPUT  
   ALBIN
The loader remains in ALBIN mode until the next block end ($) .

3. The expression *ADRES means in BIN mode: set ADRES in the location counter (origin setting) but in ALBIN mode: set the contents of the location counter in ADRES ("key-setting")
(The location counter contains the address where the next instruction will be placed.)
Due to the fact that the PAL III Assembler does not differentiate between BIN and ALBIN mode, the expression *ADRES in an ALBIN block always has to be followed by a restoring expression, which restores the location counter of the assembler, for example:

\[
\begin{align*}
\text{AA,} & \\
\text{*ADRES} & \\
\text{*AA + 1} & \\
\end{align*}
\]

The loader interprets this as a "key-setting" in AA + 1 but this is irrelevant since AA + 1 is filled with the next program instruction.

4. Each ALBIN mode block is translated beginning with an address at page 1. This means that the first restoring expression has to be: .200 + x, x is the relative page address where the program has to be placed. This expression must occur before the first instruction of the block. It may, however, be preceded by one or more "key-settings":

\[
\begin{align*}
\text{*INPUT} & \\
\text{ALBIN} & \\
\text{*SUBI} & \\
\text{*200} & \\
\text{SUB, 0} & \\
\end{align*}
\]

This program can be loaded from the beginning of each page and its actual place then has been noted down in the (page 0) address SUBI.

5. Reading a binary tape, the ALBIN loader will halt at the beginning of each ALBIN mode block. The actual starting address of the block then has to be set via the switch register. Loading will follow by pressing the continue button.

The starting address must agree with paragraph 4 which implies that the least significant 7 bits of the switch register must contain x. In most cases, x can be 0. The blocks then always start at the beginning of a memory page.
6. Sample conversion of a symbolic BIN mode program block in ALBIN mode.

```
/BIN
  KEY
*SUB1, SUB1
SUB2, SUB1

*BEG
SUB1, ø
  --
  --
  JMP I SUB1
  --
  BB, JMP I SUB1

SUB2, ø
  --
  --
  TAD I AAI
  --
AAI, AA
  --

--- --- --- --- --- --- --- --- --- --- --- "Page Transition"

  --
  --
  CC, --
  *AAI
  *CC+1

AA, NUMBER
  --

$ --

```
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7400</td>
<td>6032</td>
<td>BEGRIM, KCC</td>
</tr>
<tr>
<td>7401</td>
<td>4216</td>
<td>JMS READIN</td>
</tr>
<tr>
<td>7402</td>
<td>7106</td>
<td>CLL RTL</td>
</tr>
<tr>
<td>7403</td>
<td>7006</td>
<td>RTL</td>
</tr>
<tr>
<td>7404</td>
<td>7510</td>
<td>SPA</td>
</tr>
<tr>
<td>7405</td>
<td>5201</td>
<td>JMP BEGRIM+1</td>
</tr>
<tr>
<td>7406</td>
<td>7006</td>
<td>RTL</td>
</tr>
<tr>
<td>7407</td>
<td>6031</td>
<td>KSF</td>
</tr>
<tr>
<td>7410</td>
<td>5207</td>
<td>JMP -1</td>
</tr>
<tr>
<td>7411</td>
<td>6034</td>
<td>KRS</td>
</tr>
<tr>
<td>7412</td>
<td>7420</td>
<td>SNL</td>
</tr>
<tr>
<td>7413</td>
<td>3753</td>
<td>DCA I KOBUS</td>
</tr>
<tr>
<td>7414</td>
<td>3353</td>
<td>DCA KOBUS</td>
</tr>
<tr>
<td>7415</td>
<td>5600</td>
<td>JMP I BEGRIM</td>
</tr>
<tr>
<td>7416</td>
<td>0000</td>
<td>READIN, 0</td>
</tr>
<tr>
<td>7417</td>
<td>6031</td>
<td>KSF</td>
</tr>
<tr>
<td>7420</td>
<td>5217</td>
<td>JMP -1</td>
</tr>
<tr>
<td>7421</td>
<td>6036</td>
<td>KRB</td>
</tr>
<tr>
<td>7422</td>
<td>3346</td>
<td>DCA CHAR</td>
</tr>
<tr>
<td>7423</td>
<td>1346</td>
<td>TAD CHAR</td>
</tr>
<tr>
<td>7424</td>
<td>5616</td>
<td>JMP I READIN</td>
</tr>
<tr>
<td>7425</td>
<td>7240</td>
<td>BEGN, CLA CMA</td>
</tr>
<tr>
<td>7426</td>
<td>3344</td>
<td>DCA INDIC</td>
</tr>
<tr>
<td>7427</td>
<td>3345</td>
<td>DCA CHKSM</td>
</tr>
<tr>
<td>7430</td>
<td>3352</td>
<td>DCA FIRST</td>
</tr>
<tr>
<td>7431</td>
<td>4320</td>
<td>JMS BEGG</td>
</tr>
<tr>
<td>7432</td>
<td>5231</td>
<td>JMP -1</td>
</tr>
<tr>
<td>7433</td>
<td>1346</td>
<td>GO, TAD CHAR</td>
</tr>
<tr>
<td>7434</td>
<td>3347</td>
<td>DCA CKT</td>
</tr>
<tr>
<td>7435</td>
<td>1346</td>
<td>TAD CHAR</td>
</tr>
<tr>
<td>7436</td>
<td>7106</td>
<td>CLL RTL</td>
</tr>
<tr>
<td>7437</td>
<td>7006</td>
<td>RTL</td>
</tr>
<tr>
<td>7440</td>
<td>7006</td>
<td>RTL</td>
</tr>
<tr>
<td>7441</td>
<td>3350</td>
<td>DCA WORD</td>
</tr>
<tr>
<td>7442</td>
<td>4216</td>
<td>JMS READIN</td>
</tr>
<tr>
<td>7443</td>
<td>1350</td>
<td>TAD WORD</td>
</tr>
<tr>
<td>7444</td>
<td>3350</td>
<td>DCA WORD</td>
</tr>
<tr>
<td>7445</td>
<td>7430</td>
<td>SZL</td>
</tr>
<tr>
<td>7446</td>
<td>7240</td>
<td>CLA CMA</td>
</tr>
<tr>
<td>7447</td>
<td>3351</td>
<td>DCA MEM</td>
</tr>
<tr>
<td>7450</td>
<td>1346</td>
<td>TAD CHAR</td>
</tr>
<tr>
<td>7451</td>
<td>1347</td>
<td>TAD CKT</td>
</tr>
<tr>
<td>7452</td>
<td>3347</td>
<td>DCA CKT</td>
</tr>
<tr>
<td>7453</td>
<td>4320</td>
<td>JMS BEGG</td>
</tr>
<tr>
<td>7454</td>
<td>5357</td>
<td>JMP BEND</td>
</tr>
<tr>
<td>7455</td>
<td>2351</td>
<td>ISZ MEM</td>
</tr>
<tr>
<td>7456</td>
<td>5302</td>
<td>JMP STORE</td>
</tr>
<tr>
<td>7457</td>
<td>1344</td>
<td>TAD INDIC</td>
</tr>
<tr>
<td>7460</td>
<td>7640</td>
<td>SZA CLA</td>
</tr>
<tr>
<td>7461</td>
<td>5306</td>
<td>JMP B1</td>
</tr>
<tr>
<td>7462</td>
<td>1352</td>
<td>TAD FIRST</td>
</tr>
<tr>
<td>7463</td>
<td>7650</td>
<td>SNA CLA</td>
</tr>
<tr>
<td>7464</td>
<td>5311</td>
<td>JMP B2</td>
</tr>
<tr>
<td>7465</td>
<td>1350</td>
<td>B4, TAD WORD</td>
</tr>
</tbody>
</table>
7466 1273 TAD M275
7467 7516 SPA
7468 5273 JMP B3
7471 1353 TAD KOBUS
7472 3345 DCA WORD
7473 7657 \b3, M280, 7657 /CLA
7474 1354 TAD ORIGIN
7475 3750 DCA I WORD
7476 1347 CHEX, TAD CKT
7477 1345 TAD CHKSM
7566 3345 DCA CHKSM
7591 5233 JMP GO
7522 1350 STORE, TAD WORD
7563 3754 DCA I ORIGIN
7544 2354 ISZ ORIGIN
7505 5276 JMP CHEX
7566 1356 \b1, TAD WORD
7567 3354 DCA ORIGIN
7512 5276 JMP CHEX
7511 2352 \b2, ISZ FIRST
7512 7402 HLT
7513 7624 CLA OSR
7514 3353 DCA KOBUS
7515 1353 TAD KOBUS
7516 3354 DCA ORIGIN
7517 5265 JMP B4
7522 0350 BEGG, \b
7521 3355 DCA SWITCH
7522 4216 JMS READIN
7523 1356 TAD M377
7524 7644 SZA CLA
7525 5331 JMP .+4
7526 2355 ISZ SWITCH
7527 7244 CLA CMA
7530 5321 JMP BEGG+1
7531 1355 TAD SWITCH
7532 7644 SZA CLA
7533 5322 JMP BEGG+2
7534 1346 TAD CHAR
7535 3341 AND MASK
7536 1273 TAD M275
7537 745C SNA
7540 5723 JMP I BEGG
7541 7700 MASK, SNA CLA
7542 5322 JMP BEGG+2
7543 2320 ISZ BEGG
7544 JED INDIC, \b
7545 0000 CHKS3, \b
7546 0000 CHAR, \b
7547 0000 CKT, \b
7550 0000 WORD, \b