VAX/VMS
Training

VAX/VMS
Device Driver
Driver Incorporation

digital
## CONTENTS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>4-3</td>
</tr>
<tr>
<td>OBJECTIVES</td>
<td>4-3</td>
</tr>
<tr>
<td>RESOURCES</td>
<td>4-3</td>
</tr>
<tr>
<td>TOPICS</td>
<td>4-5</td>
</tr>
<tr>
<td>ASSEMBLING A DRIVER</td>
<td>4-7</td>
</tr>
<tr>
<td>LINKING A DRIVER</td>
<td>4-8</td>
</tr>
<tr>
<td>INTEGRATING A DRIVER WITH THE SYSTEM</td>
<td>4-9</td>
</tr>
<tr>
<td>COMMANDS</td>
<td>4-10</td>
</tr>
<tr>
<td>CONNECT QUALIFIERS</td>
<td>4-11</td>
</tr>
<tr>
<td>The SHOW Command</td>
<td>4-12</td>
</tr>
</tbody>
</table>

## EXAMPLES

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>SYSGEN SHOW/DEVICES Command</td>
<td>4-13</td>
</tr>
<tr>
<td>4-2</td>
<td>SYSGEN SHOW/CONFIG Command</td>
<td>4-14</td>
</tr>
<tr>
<td>4-3</td>
<td>SYSGEN Command vs. Driver Incorporation</td>
<td>4-15</td>
</tr>
</tbody>
</table>
INTRODUCTION

After a driver is written, it must be assembled, linked, and integrated (loaded) with the system for testing, and final operation. A driver can be loaded into the system any time after the system is bootstrapped. If the driver contains an error which does not crash the system, the error may be corrected, and a new version of the driver reloaded. This module examines the commands and procedures for assembling, linking, and loading drivers.

OBJECTIVES

Upon completion of this module, you will be able to:

1. Assemble and link a user-written driver at a terminal.

2. Use the SYSGEN commands LOAD, CONNECT, SHOW and RELOAD to integrate a user-written driver with the system.

RESOURCES

1. Guide to Writing a Device Driver for VAX/VMS

2. PDP-11 Peripherals Handbook

TOPICS

- Creating an Executable Driver
  - Assembling
  - Linking
- Integrating a Driver with VMS
  - SYSGEN Utility
  - SYSGEN Commands
  - Examples
ASSEMBLING A DRIVER

$MACRO/LIST full-driver-file-spec+SYS$LIBRARY:LIB/LIB

Example:

$MACRO/LIST DB0:[XDIR]XXDRIVER+SYS$LIBRARY:LIB/LIB

• Since drivers invariably invoke many system macros, it is essential that the system macro library (SYS$LIBRARY:LIB/LIB) be assembled with drivers.

• The listing file (xxDRIVER.LIS) is essential when debugging the driver.
LINKING A DRIVER

Format:

```
MYDRIVER.OPT
BASE=0
```

```
$ LINK/NOTRACE/MAP/FULL XXDRIVER1 [,XXDRIVER2,....],-
$ _MYDRIVER/OPTIONS,-
$ _SYSSYSTEM:SYS.STB/SELECTIVE
```

- The DCL continuation mark (\-\) is used here for convenience only.
- MYDRIVER.OPT is an option file that specifies a zero base for the executable image.
- If the driver consists of multiple source files, XXDRIVER1 must contain the driver prologue table.
- The image file created is XXDRIVER1.EXE. The map file created is XXDRIVER1.MAP.
- It is normal to see the following message displayed at LINK time: "XXDRIVER1 has no user transfer address".
- The map file is essential when debugging the driver.
- If the /SELECTIVE qualifier is omitted, the entire system symbol table is included (which makes for extremely large map files).
- It is suggested that you write a command procedure to assemble and link your driver.
INTEGRATING A DRIVER WITH THE SYSTEM

$ RUN SYSSSYSTEM:SYSGEN
SYSGEN> COMMAND
SYSGEN> EXIT

<table>
<thead>
<tr>
<th>Function</th>
<th>Command</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load driver code into nonpaged pool</td>
<td>LOAD</td>
<td>Load DBA0:[MARSH]mydriver.exe</td>
</tr>
<tr>
<td>Create I/O data structure(s)</td>
<td>CONNECT</td>
<td>CONNECT LPA0/qualifiers</td>
</tr>
<tr>
<td>Replace a driver with another</td>
<td>RELOAD</td>
<td>RELOAD LPDRIVER</td>
</tr>
<tr>
<td>Display information</td>
<td>SHOW/qualifier</td>
<td></td>
</tr>
<tr>
<td>Devices</td>
<td>/DEVICES</td>
<td>SHOW/DEVICES [=DB]</td>
</tr>
<tr>
<td>Configuration</td>
<td>/CONFIGURATION</td>
<td></td>
</tr>
</tbody>
</table>

Privilege Requirements vs Commands

CMKRNRL: Load, Connect, Reload

CMEXEC: SHOW
DRIVER INCORPORATION

COMMANDS

LOAD

LOAD driver_filespec

- Allocates nonpaged pool
- Bring driver image into memory
- Unload old driver image if existent and if possible

Defaults:   Directory = SYS$SYSTEM
            File_type = Exe

CONNECT

- Load driver, if not already in memory
- Allocate nonpaged pool
- Build I/O data base
  - If first unit on controller create UCB, IDB, DDB
  - If not first unit, create UCB only

Example:

SYSGEN>CONNECT LPA0/Adapter=3/CSR=%0777514/VECTOR=%0200

Defaults:   /DRIVER=LPDRIVER
            /NUMVEC=1
            /ADPUNIT=0

DEVICE NAME

devcu

devc = device type

c = alphabetic controller designation

u = unit number (0-7)

Dev and c must specify a unique and accurate combination. If the specified combination already exists, no new control blocks are created. If not accurately specified, spurious blocks may be created.

Paper Tape Punch Example

SYSGEN>LOAD [DRIVER1]PPDRIVER.EXE
SYSGEN>CONNECT DPA0/CSR=%/0777554/VECTOR=%074/ADAPTER=UB0
**CONNECT QUALIFIERS**

<table>
<thead>
<tr>
<th>Qualifier and Purpose</th>
<th>Usage/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Adapter = tr_number</td>
<td>Required for connecting all units</td>
</tr>
<tr>
<td>/VEC = vector_addr</td>
<td>Required for connecting first unit</td>
</tr>
<tr>
<td>/NUMVEC = # of interrupt vectors</td>
<td>Required for connecting first unit</td>
</tr>
<tr>
<td>/CSR = UNIBUS addr of CSR</td>
<td>Required for connecting first unit</td>
</tr>
<tr>
<td>/DRIVER = name (as specified in DPT)</td>
<td>Default = aa driver (aa is device type)</td>
</tr>
<tr>
<td>/ADPUNIT = unit # of device on controller (MASSBUS only)</td>
<td>Default = 0</td>
</tr>
<tr>
<td>/MAXUNITS = # (Max units/controller)</td>
<td>Default = 8 or value in DPTAB macro (first unit only)</td>
</tr>
</tbody>
</table>

**RELOAD**

RELOAD driver_filespec

Same as LOAD except no check is made to see if driver_filespec = driver_name is list of DPTs.

If DPT$M NOUNLOAD bit is set in flags parameter to DPTAB macro, driver cannot be reloaded.

Can be used to recover nonpaged pool by RELOADing a large infrequently used driver with a dummy driver. However, the best way to get rid of a driver is to reboot.
The SHOW Command

Displays the values of system parameters in the SYSGEN work area, plus the I/O driver database.

Format:

SHOW parameter-name
SHOW /qualifier

Information
Displayed

Devices by name, number of units, nexus number, adapter type, and CSR and vector addresses.

Displays the device data base.

The beginning and ending addresses of the drivers.

Qualifier

/CONFIGURATION
(CMEXEC privilege required, see note below)

/DEVICE[=device-driver]
(CMEXEC privilege required)

/DRIVER[=device-driver]

NOTE
Also accepts /ADAPTER = adapter-spec, /OUTPUT = filespec, as well as /COMMAND_FILE, which formats all the devices as CONNECT commands to allow a complete reconfiguration of a system's UNIBUS without the use of AUTOCONFIGURE.
Example 4-1  SYSGEN SHOW/DEVICES Command

<table>
<thead>
<tr>
<th>Output</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVER</td>
<td>Driver name</td>
</tr>
<tr>
<td>START</td>
<td>Starting virtual address of DPT</td>
</tr>
<tr>
<td>END</td>
<td>Ending virtual address of driver</td>
</tr>
<tr>
<td>DEV</td>
<td>Generic device/controller name</td>
</tr>
<tr>
<td>DDB</td>
<td>Virtual address of controller data structures</td>
</tr>
<tr>
<td>CRB</td>
<td></td>
</tr>
<tr>
<td>IDB</td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td>Unit number of device</td>
</tr>
<tr>
<td>UCB</td>
<td>Virtual address of Unit Control Block</td>
</tr>
</tbody>
</table>
$RUN SYS$SYSTEM:SYSGEN

SYSGEN> SHOW/CONFIG

System CSR and Vectors on 2-JUL-1982 12:35:41.94

Name: DRA Units: 2 Nexus:4 (MBA)
Name: LPA Units: 1 Nexus:8 (UBA) CSR: 777514 Vector1: 200 Vector2: 000
Name: TTA Units: 8 Nexus:8 (UBA) CSR: 760100 Vector1: 300 Vector2: 304

SYSGEN> SHOW/CONFIG/COMMAND
% RUN SYS$SYSTEM:SYSGEN
AUTOCONFIGURE 4
CONNECT LPA0 /ADAP=B /CSR=0777514 /VECT=0200 /NUMV=01 /DRIVER=LPDRIVER
CONNECT TTA0 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA1 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA2 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA3 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA4 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA5 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA6 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
CONNECT TTA7 /ADAP=B /CSR=0760100 /VECT=0300 /NUMV=02 /DRIVER=DZDRIVER
EXIT

Example 4-2 SYSGEN SHOW/CONFIG Command
Example 4-3  SYSGEN Commands vs. Driver Incorporation