Table of Contents

Introduction

This manual describes a set of macros designed for use with the Hercules emulator.

The following macros found in the maclib macro library directory belong to the herc category.

Macro	Description
HRCCMD	Issue a console command from the program to the Hercules emulator
HRCIPLP	Save the parameters provided to the program by the Hercules ipl command.

How to Use This Document

If this is the first time you are reading this document, it is recommended that you read the "HOWTO..." section of interest first. This gives the background helpful in understanding where each macro is used with the Hercules emulator.

For information related to architecture levels, refer to either SATK.odt or SATK.pdf. Architecture levels apply to nearly all macros.

HRCCMD

Source file: macros/HRCCMD.mac

Macro Format:

```
[label]    HRCCMD cmd,
                  [,CMDL=r]
                  [,FAIL=label]
```

The HRCCMD macro allows the program to issue a console command to the Hercules emulator on which the program is executing.

Before using this macro, it is strongly recommended that the section “HOWTO Issue a Hercules Command from a Program” be reviewed. There is information in this HOWTO section that is not well documented or easily found within the Hercules web pages or in other locations.

Assembly Considerations:

- A preceding ARCHLVL macro is required.
- A local base register must be established preceding HRCCMD.

Execution Consideration:

- A command response buffer is not supplied by HRCCMD.
- A command length of zero causes the Hercules emulator to place the executing CPU in the stop state.

Label Field Usage:

The label field, if present, defines an assembly label associated with the first instruction generated by the macro.

Positional Parameters:

1. The required cmd positional parameter identifies the register pointing to the EBCDIC command string being passed to the Hercules emulator for execution.

Keyword Parameters:

Keyword	Default	Description
CMDL	none	Specifies the register containing the length of the command in bytes.

Keyword	Default	Description
		If omitted, the assumption is made that the cmd register points to a halfword immediately preceding the EBCDIC command that contains the command's length. If omitted, the command length is placed in the cmd positional parameter plus 2.
FAIL	none	Identifies the label to which control is passed if the command fails. If omitted, a failure will cause a disabled wait PSW to be loaded as the active PSW with an instruction address of X'400'.

Programming Note:

This macro allows the program to influence how the Hercules emulator operates. Program and I/O tracing can be initiated or terminated by the program. Devices can be attached or detached.

Messages may also sent by the program to the console for display using the Hercules msg or msgnoh console commands.

HRCIPLP

Source file: macros/HRCIPLP.mac

Macro Format:

```
[label]    HRCIPLP [ADDR=[X'200'|addr]]
                   [,CLEAR=[YES|NO]]
```

The Hercules emulator can provide to the program parameters when supplied by the user in the ipl console command. Up to 64 EBCDIC characters may be communicated as IPL parameter information. The parameter information, when specified, is placed contiguously within the 32-bit registers as EBCDIC characters starting with general register 0 and proceeding through to general register 15 if required. The HRCIPLP macro stores the parameter information in memory for later program inspection and use.

If the user's ipl command does not include any parameters, all registers are reset to zero as is expected during the CPU reset occurring during the IPL function. HRCIPLP stores the zero register content in this case at the designated location.

Assembly Considerations: None

Execution Considerations:

- The IPL parameter information may be in lieu of or in addition to any content supplied by the Hercules LOADPARM configuration statement. HRCIPLP does not access LOADPARM data.
- Register content must not be changed prior to executing HRCIPLP. Changing register content may corrupt the parameter information before it is preserved in memory.
- A base register must not be established preceding HRCIPLP. See the preceding consideration
- When the Hercules IPL parameters are stored, the label associated with HRCIPLP must be the bare-metal program's entry location in its IPL PSW. See the second consideration.

Label Field Usage:

The label field, if present, defines an assembly label associated with the first instruction generated by the macro.

Positional Parameters: None

Keyword Parameters:

Keyword	Default	Description
ADDR	X'200'	Specifies the absolute address within the first 4096 bytes of memory at which the 64 bytes of parameter information is stored.
CLEAR	NO	Specifies whether the registers containing the IPL parameters should be reset to zero or not: • YES – clears all 32-bit IPL registers to zero. • NO – leaves the IPL parameter information unchanged in the 32-bit registers.

HOWTO Issue a Hercules Command From a Program

The Hercules emulator implements a DIAGNOSE instruction similar to that provided by the VM hypervisor's Virtual Console Function.

The program uses this instruction to issue a console command to the Hercules emulator:

```
DIAGNOSE R1,R2,X'008'
```

R1 designates a register containing the real address of the command being issued.

R2 designates the register containing the length of the command, which may not exceed 255 bytes.

R2, in bits 0-7 (of a 32-bit register) or bits 32-39 (of a 64-bit register) also contains a set of flags. Hercules allows flags X'80', X'40', and X'20', but only processes flag X'40'.

When flag X'40' is set in R2, a response buffer has been supplied by the program. In this case register R1+1 contains the real address of the command response buffer and R2+1 contains the length of the response buffer. If a response buffer is supplied neither R1 nor R2 may be 15 nor may the register pairs overlap. If R1 and R2 are the same, the command area is used for the command response.

If the use of DIAGNOSE X'008' is disabled by the DIAG8CMD configuration statement or there is a problem with the command length or registers (when flag X'40' is set), a specification exception occurs.

Condition code 0 is returned upon execution of the command and reception of the response, when requested. Condition code 0 does not mean that the command was successful only that the command was issued. The command might be invalid and fails.

When a response is requested and condition code 0 is set, register R2+1 contains the length of the stored response.

If the requested response exceeds the length of the response area:

- the condition code is set to 1 and
- register R2+1 contains the number of response bytes that could not be stored in the response area.

Register R2 contains a return code, which is always set to 0 by Hercules.

The `HRCCMD` macro sets up the registers for a command without a response.

Hercules Related Configuration Statements

Two configuration statements influence how Hercules processes DIAGNOSE X'008'

1. DIAG8CMD
2. SHCMDOPT

DIAG8CMD controls whether the DIAGNOSE X'008' is allowed. Its first parameter is either enable or disable (the default). When DIAGNOSE X'008' is disabled, attempt to use it causes a specification exception. When using HRCCMD, DIAG8CMD must be set to enable.

SHCMDOPT when used disables use of the Hercules shell command. To issue the shell command (sh) from a program, the SHCMDOPT statement must **not** be present in the configuration file.

A common practice is to use the Hercules msg or msgnoh commands to issue a message to the console from the program without the need for input/output support of a console. In this case the DIAG8CMD must specify NOECHO for its second parameter.

Sending A Message to the Console

The Hercules msg or msgnoh commands may be used to send a message to the Hercules console (and correspondingly log it). These two commands are modeled after similar commands supplied by the VM hypervisor for sending messages to other users.

The syntax of the message commands are:

```
msg dest message text...

msgnoh dest message text...
```

Hercules requires the dest argument but does not process it. An asterisk is used to send a message to the local console. This is consistent with Hercules action, and only takes one character, so is typically used to send a message using DIAGNOSE X'008' to the console.

The entire command is assembled into the program like this:

```
HELLO    DC    C'msgnoh * Hello World!'
HELLOLEN EQU   *-HELLO
```

The R1 register will point to HELLO and the R2 register will contain the length calculated by HELLOLEN.

This illustrates using HRCCMD to display the message on the console:

```
LA    1,HELLO
L     2,=A(HELLOLEN)
HRCCMD 1,CMDL=2
```

And will result in this line on the Hercules console:

```
Hello World!
```

If the command is changed to msg from msgnoh, the text will be preceded in the log with date, time and additional information when displayed.

HOWTO Communicate with the Program

The preceding section explains how the program can communicate with the user using Hercules provided facilities. This section explains how the user can communicate information to the bare-metal program when it is loaded into storage.

While there are a number of ways to start a bare-metal program, the most common is to perform the Initial Program Load (IPL) function, bringing the program into storage and passing control to it.

In the typical case, media content is prepared on a Hercules emulated device allowing the media to participate in the IPL function. SATK supports that approach for ASMA developed programs using the `iplasma.py` tool. Currently the tool is limited to a Fixed-Block-Architecture (FBA) Direct Access Storage Device (DASD). The device is defined in the Hercules configuration file with a device identifier. The device is then the target of the Hercules `ipl` console command.

A less common approach is to place one or more portions of the program into binary image files and use the list-directed IPL function to load the pieces into storage and pass control to the program. The SATK assembler, A Small Mainframe Assembler (ASMA), directly supports this approach through its `--gldipl` output option. This output option creates the control file that directs the loading process and the loaded pieces of image content. The target of the Hercules `ipl` console command is the control file.

In both scenarios the Hercules `ipl` or `iplc` console command is used to load the program and pass control to it. The differences between the two commands is that `iplc` clears memory before loading the program and `ipl` does not.

Using IPL Parameters

The Hercules command syntax for passing parameters to the program is:

```
ipl[c] <device|file> [ parm word... ]
```

The characters in each `word` are converted to upper case EBCDIC with a single EBCDIC space separating each word. A maximum of 64 characters (after unnecessary spaces are removed) are placed within general registers 0-15 as if they were loaded by a LOAD MULTIPLE instruction. The first four characters are placed in general register 0, the next four in general register 1, etc. When the number of characters is less than 64, the remaining bytes

in the general register(s) remain zero.

The HRCIPLP macro preserves the parameters as supplied by Hercules for later use by the program. The program can then inspect the parameter characters and respond as implemented by the program.

The primary advantage of using the parm option is that it is supported by all Hercules supported architectures. The alternative option, loadparm, requires use of the service processor via the SERVC instruction that is not available in Hercules System/370 mode.