The Helios File Server
Installation Information

PERIHELION SOFTWARE LTD

November 1990

Copyright

This manual is Edition 2.0, November 1990.

Printed in the UK.
PDF generated in AT (Vienna).

Acknowledgements

The Helios File Server was written by N.Garnett, C.Selwyn, A.Ishan, H.Ermen and J. Powell. Helios software is available for multi-processor systems hosted by a wide range of computer types. Information on how to obtain copies of Helios software is available from Distributed Software Limited, The Maltings, Charlton Road, Shepton Mallet, Somerset BA4 5QE, UK. (Telephone: 0749 344345.)

Contents

Chapter 1

Introduction

These notes explain how to configure the file server for different disc formats and hardware controllers. You should already have followed the installation instructions in Chapter 2 of 'The Helios File Server'. Helios supports most leading makes of transputer or SCSI interface TRAMs and boards. It also runs on a PC hard disc partition. Several possible configurations are currently available for the Helios File Server, and their individual hardware requirements are described on the following pages.

Chapter 2

Device support

Helios provides a device driver support between the file server and the controller hardware. Drivers are provided to support most of the leading makes of interface. Device drivers are located in the `/helios/lib` directory. The conventional format for a Helios device driver is: `<name>.d`

2.1 The devinfo file

Before you run the file server, make an appropriate entry in the `devinfo.src` file. The example file contains entries for each of the currently supported devices. Find the discdevice which corresponds to the device you want to use, and edit the drive parameters to match your disc. You must also decide how much cache memory the file server should be allowed to use. The practical minimum is 100k, and any value of over 1 Mbyte will allow most of your working files to be cached. When you have edited the `devinfo.src` file, you should compile it again by using the following command:

```
gdi /helios/etc/devinfo.src /helios/etc/devinfo
```

Note: For a detailed description of the `devinfo.src` file, see the Appendix.

2.2 The Parsytec MSC

Device driver: `msc.d`

This uses the Parsytec MSC SCSI interface board, which offers 4 Mbytes of RAM, an SCSI interface and a T800 processor. The Helios File Server runs under Helios on this processor, and it uses most of the RAM for its own cache. A suitable SCSI drive with a capacity of up to 2 Gbytes should be added.

2.3 The Hunt HE1000

Device driver: `he1000.d`

This version uses the Hunt Engineering HE1000 SCSI interface TRAM. It has a 16 bit transputer, and controller software from ROM. It uses a link interface with which the Helios File Server communicates. HFS must run on an adjacent 32 bit processor under Helios. The device driver can communicate with the HE1000 by using any one of the four links which are available.

2.4 The Transtech TTM11-2

Device driver: `he1000.d`

The HE1000 is also supplied by Transtech, as the TTM11-2.

2.5 The T2SL Systems Paradise-1

Device driver: `paradise.d`

In this configuration, HFS runs under Helios on one of the existing network transputers. This is connected through an Inmos serial link to a T2 Systems Paradise-1 SCSI transputer module. The Paradise-1 is based on a T222 transputer, controlled by firmware. It can be coupled to any SCSI Winchester disc drive of up to 3 Gbytes. The firmware manages the low level SCSI interface. This driver is supplied on request from T2 Systems, Ltd.

2.6 The Inmos IMS B422

Device driver: `b422.d`

This version uses the Inmos B422 SCSI TRAM. It uses a 20 MHZ 16 bit transputer with 64k of S-ram. Controller software is downloaded (file b422.btu) from the host.

2.7 The Paratech GM8504

Details of this driver can be obtained directly from Paratech.

2.8 Rawdisc support

This version uses any spare partition on the hard disc of a PC host. The Helios file server can run on any transputer which has sufficient memory available for the cache, but the further the transputer is away from the host, the longer the disc access delays will be. A spare partition needs to contain a minimum of 20 Mbytes to be of use. A pseudo tape device is supported. It uses the file `/helios/tmp/tape` if the tape volume is defined in the `devinfo` file.

2.9 The A.G. Electronics M200

Device driver: `m212.d`

This controller uses the M212 disc controller by Inmos, which provides a 16 bit processor. The Helios File Server must be run on an adjacent 32 bit processor under Helios. The A.G. Electronics M200 hard disc system has a device interface which can use the multi sector operations supported by the firmware for this device.

2.10 The Transtech TMB05

Device driver: `m212.d`

The Transtech TMB05 is compatible with the A.G. Electronics M200 hard disc system.

2.11 PC disc partition

The Helios file server can use a spare partition on a PC hard disc. This allows the restrictions of filenames to be lifted, and introduces performance benefits through the caching of the filing system.

2.12 The contents of the release disc

The files contained in the release disc are listed in table 2.1.

lib	-- **Server and Drivers**
	-- the file server(fs)
	-- device drivers for disc interface hardware
etc	-- **Device Information**
	-- example devinfo source (devinfo.src)
	-- compiled version (devinfo)
bin	-- **Commands**
	-- display capabilities (access)
	-- file protection (chmod)
	-- disc editor (de)
	-- format partition (fsformat)
	-- alter sync mode (fsync)
	-- devinfo compiler (gdi)
	-- load M212 firmware (loadm2)
	-- capability manipulation (refine)
	-- flush cache to disc (sync)
	-- terminate file server (termfs)
devs	-- **Driver Sources**
msdos	-- **MSDOS Partition Builder** (makedisk.exe)

Table **2.1** The contents of the file server release disc.

Chapter 3

Installation: hardware-specific entries

Device drivers obtain disc characteristics from the `devinfo` file. (An example of this file is supplied with the release disc.) Before using the file server, you should edit the `devinfo.src` file to match the characteristics of your disc, and then recompile it with the `gdi` command. (See Appendix A.1 for an example.)

3.1 Raw disc

The raw disc service is provided by a server within the Helios I/O server, and uses a spare partition on the host machine. This provides you with a directory called `/rawdisc`, containing a list of partitions which are available for the file server to use. You will need to initialise the parameters in the fileserver and discdevice structures of the `devinfo.src` file, to set up the partitions you intend to use. Place the number of the partition to be used in the controller field of the discdevice entry. Run the makedisk program with the partition letter as its only argument, and the disc parameters, (such as sectorsize, and the number of sectors, tracks and cylinders),will be displayed on your screen. Copy the entries into the `devinfo.src` file. If you have already run the makedisk program once, simply rerun it to obtain this information.

3.2 The Parsytec MSC

The board should be marked as a system processor in the resource map, and the cache should use about 75 percent of the available memory. The file server MSC should contain entries for **cgsize** (blocks per cylinder group), and **ncg** (number of cylinder groups) The **ncg** rating should be in the range 32 to 64. The product of both values multiplied by the block size, (which is 4Kbytes), gives the capacity of the disc, so the drive entry in the disc device MSC is not used.

3.3 The A.G. Electronics M200

The M212 device requires the disc parameters to be defined *exactly*, so that it can initialise the controller. You will also need to give the number of the link which provides your access to M212. You should place the link number in the **controller** field of the disc device entry. If you want to use the A.G. Electronics firmware, you should set the disc device **mode** to indicate the buffering that you intend to use. The bottom 4 bits will indicate the 'read' mode, and the next 4 bits will indicate the 'write' mode. The default value, which uses M212 commands only, is 0 (Zero). A value of 1 would use 'buffered' mode, and a value of 2 would use 'unbuffered' mode. You can use different modes for 'read' and 'write' operations. Remember that you will need to load the A.G. Electronics board with its controlling software before you can run the file server or the disc editor. You can load the software in two different ways. (1) Use the software supplied by A.G. Electronics and load from the board's PC link. (2) Load from Helios, by using the `loadm2` command. In either case, you will only need to load the software once, unless the M212 is reset for some reason.

3.4 The Hunt HE1000

The disc parameters for the HE1000 entry in the `devinfo` file must be updated to correspond with the disc you are using. You must also give the number of the link you are using to access the device. You should place the link number in the **controller** field of the disc device entry. The SCSI id of the disc should be placed in the **id** field, and that of the HE1000 should be placed in the **mode** field. The drive fields should also give the sectorsize, sectors, tracks and cylinder values for your disc. (There is an example in the `devinfo.src` file under the name `he1000` for both the fileserver and discdevice entries.) These parameters will be used by the `fsformat` command and the HFS drivers.

3.5 The Transtech TTM11-2

The HE1000 is supplied by Transtech as the TTM11-2, so the installation entries are the same as those for the HE1000, as described in the previous section.

3.6 The T2SL Paradise-1

The disc parameters for the Paradise-1 TRAM must be entered in the devinfo.src file. (There is an example in the devinfo.src file under the name `paradise` for both the fileserver and discdevice entries.) The fileserver must be run on a processor which is adjacent to the Paradise-1 TRAM. The example devinfo entry for the Paradise-1 was written for a 40 Mbyte disc drive. The user must update the cylinder group size (cgsize), and the number of cylinder groups (ncg), parameters to reflect the actual capacity of the user's disc drive. The total disc capacity (in bytes), available to the file server will be `4096 x cgsize x ncg`. This should be less than (or equal to) the actual disc capacity.

The controller field and the id field will also need to be updated. The controller field defines which link of the HFS processor connects to the Paradise-1, and the id field defines the SCSI id of the disc drive.

3.7 The Inmos IMS B422

The disc parameters for the IMS B422 TRAM must be entered in the devinfo.src file. The controlling link is specified in the controller field, and the SCSI id of the disc is specified in the id field. The sectorsize, sectors, tracks and cylinder fields must correspond to the disc.

Chapter 4

Formatting and initialisation

Before you can run the file server on a disc or on a partition, you must first format the medium, and then initialise the medium with the file server data structures. The steps you must take to do this will vary according to the type of machine you are using.

1. The MSC disc is already formatted.
2. You can format an M212 disc, a Hunt SCSI disc or a T2SL Paradise-1 by using `fsformat`. Other manufacturers may supply their own formatting software.
3. PC discs should already be formatted. (If not, you can format them by using the relevant software on the host machine.) You will also need to initialise the partition as an MSDOS partition, and then you must run the program `makedisk.exe`, with the partition letter as its only argument. This will make MSDOS think that the partition is full, so that it will not try to use it.

When you have formatted the drive or the partition, you must initialise it with the data structures of the Helios file server. To do this, you use the command:

```
/helios/lib/fs -m <name>
```

(Remember that the <`name`> on the end of that command must be the same as the name of the 'fileserver' entry in the `devinfo.src` file. It will also be the name you intend to associate with the drive or the partition you have just formatted).

Appendix A

The devinfo.src file

A.1 Compiling devinfo.src

For the information in `devinfo.src` to be made available to devices and servers, it must be compiled into a binary form. This is done by the program `gdi` (Generate Device Info), which runs as follows.

```
gdi /helios/etc/devinfo.src /helios/etc/devinfo
```

A.2 Syntax

The syntax of the `devinfo` file is as follows.

```
<devinfo>     ::=<entry>{<entry>}
<entry>       ::=<entrykey> <name> <description>
<entrykey>    ::=fileserver | discdevice | serialserver | eventserver
<description> ::={ {keyvalue} }
<keyvalue>    ::=<keyword> <value>
<value>       ::=<number> | <name> | <filename> | <description>
<number>      ::=[-][<base>]<digit>{<digit>}
<base>        ::=0[x|X]
<digit>       ::=0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F|a|b|c|d|e|f
<name>        ::=<alpha>{<alpha>|<digit>|<period>}
<alpha>       ::=(A-Z)|(a-z)
<period>      ::=.
<filename>    ::=[/]<name>{/<name>}
```

A.3 Example devinfo file

The following is an annotated example of a `devinfo` file.

```
filesrver fs
{
      device          m212  # discdevice to use
      cachesize       100   # approx cache size in k
```

```
      syncop          1     # synchronous operations

      volume {              # define a volume
        name         fs1    # volume name
        partition    0      # maps to partition 0 of disc device
      }
  volume {
     name            fs2
     partition       1      # multi partition volume
     partition       2
  }
}

fileserver raw                # Rawdisc based file system
{
        device     raw
        cachesize  100
#       syncop     1
        volume
        {
               name        raw
               partition   0
        }
        volume
        {
               name        tape
               partition   1
               type        raw
        }
}

discdevice m212
{
     name            m212.dev # device name in /helios/lib
     controller      3        # through link 3
     addressing      1        # addresses are in bytes
     mode            0x11     # MULTI buffered read and write

# partitions...
partition {                   # partition 0
   drive             0      # partition is on drive 0
   start             2      # starts at cylinder 2
                            # ends at last cylinder of drive
}
partition {                   # partition 1
   drive             1      # occupies whole drive
}
```

```
partition {                 # partition 2
   drive             2      # occupies whole drive
}

# disc drives...
drive {                     # define a physical disc drive
   id                1      # id within controller
   type              1      # type in controller
   sectorsize        512    # size of sectors in bytes
   sectors           17     # sectors per track
   tracks            4      # tracks per cylinder
   cylinders         612    # cylinders
}
 drive {                    # drive 1
    id               2

    type             1
    sectorsize       512
    sectors          17
    tracks           6
    cylinders        1034
  }

  drive {                       # drive 2
     id               2
     type             1
     sectorsize       512
     sectors          17
     tracks           6
     cylinders        1034
  }
}

serialserver link
{
   name      links       # server name
   device    link.dev    # device

   line {                # line 0
      name   link1       # addressed by /links/link1
      offset 1           # offset = link number
   }

   line {                # line 1
      name   link2
      offset 2
   }
```

```
}
```

This shows a file server which contains two volumes. **fs1** occupies just one partition and **fs2** occupies two partitions. Within the discdevice the partitions are mapped into the physical discs. The other entry is a serialserver which presents a serial stream interface to a transputer's links.

Index