Gesellschaft für
Parallele Systemtechnik mbH

parsytec

MultiCluster Series

Software Documentation

Author

A. Joachimsmeyer

HELIOS 1.1

Delivery Manual
September 1989

SUN-3/4 installation of Helios

The transputer software for SUN-UNIX computers is distributed on one 1/4" streamer-tape. To install this software on your system you must have superuser permission. After the installation every normal user may call Helios.

Be very careful during the installation of Helios. Though UNIX is a user friendly and save operating system it is very easy for the superuser to destroy a great amount of software.

Contact your local system manager for further help before starting dangerous experiments !

The installation consists of several steps:

- Installation of the hardware.
- Installation of the software.
- Finally check your installation by calling Helios.

Installation of the hardware

In order to use the transputer board it must be properly installed on your computer. The SUN must have a VME-bus and you need a VME-adaptor from SUN. First we will describe the mounting of the transputer board into the VME-adaptor frame. Afterwards the installation of the adaptor-frame in the SUN will be our aim.

The installation of the hardware requires the computer to be powered down. Power down the system. But be carefully :

Hint:

If you have any doubts about knowing exactly how to power down the system, ask your system manager. The filesystem may be damaged if you do not obey the correct powerdown procedure!

- Power down your system. Be sure that no other users are logged in or will be disturbed by your power down. This may happen if this system is used as a file server. Usually the command

  sysdown or **shutdown**

  will work fine. You are asked for confirming the shutdown and then for the shutdown time. Enter: **now**. After some seconds the message

  unix halted

  should apear. Otherwise contact your local system manager. Now you can switch off the power-supply of the computer. Remove the power cable from the computer for security reasons.

- o Select a slot of the VME-bus for the bus bridgehead board. Keep in mind that you must connect a BBK-V1 with an other transputer board via a link cable (e.g. LNK-10). Even with a BBK-V2 or VMTM it may be desirable to transfer data to other transputers using the **transfer utility**. If there are any other boards directly to the left or right hand side of the bus bridgehead board, you must do mechanical work to the adapter frame in order to let the link cable pass by.

  Only for BBK-V1 and BBK-V2 if used as bus masters: Take also in account that the bus arbitration is done using a daisy chain. The nearer the board to slot 1 (the arbiter slot) the higher its precedence.

The next two paragraphs apply to BBK-V1 and BBK-V2 only. If you are using a VMTM skip them.

Important Note:

If you are using a BBK-V1 or BBK-V2 board do remove the two jumpers described below (INT and BG3) anyway.

Though this seems neccessary only when using these boards as bus masters or interrupt server/requesters their VME-controller chip will get confused if these jumpers are present. You will be very surprised about the strange behaviour of your computer!

If you are using a VMTM the two jumpers must stay installed because the VMTM board as a pure passive bus component did not connect the daisy chains for bus-grant (BG0..3) or interrupt acknowledge (INT).

- o Dismount the cover plates to open the inner backplane cover. Have a look at the backplane jumpers of the VME-bus. For each slot you will see 4 possible jumper positions. On top there are three jumpers near to each other and one seperated by about one inch from the others.

If present remove jumpers from two positions:

- o The one standing apart from the group of three (INT).
- o From the group of three the one nearest to the lonesome jumper (BG3).

- o Make sure that both daisy chains from slot 1 up to the selected slot are not broken anyway. This means that every slot between the choosen one and slot 1 has either the two jumpers installed or contains a board (maybe disk-controllers, memory extensions etc.). The recommend slot 3 on a diskless SUN 3-140 is factory configured this way. Therefor we recommended this slot. Any other already installed boards will have the described jumper settings, especially the CPU-board in slot 1.

From here on all boardtypes are concerned.

- o Choose an available address at the VME-bus. This requires the knowledge of the address-layout on your computer. If there is no additional hardware inside the SUN you can use the default addresses as base address for the transputer board. This will require no further action on any board, because this is the default factory setting.

The default address depends on the board type. See the list below:

board-type	**default address (hexadecimal)**
BBK-V1	$FFA00000 (32 addressbits) or $A00000 (24 addressbits)
BBK-V2	$80000000
VMTM	$80000000

If the choosen address is different from the default setting notice it for further reference during the software installation.

- o Make sure that the jumper settings are the default factory settings. They are described in the appropriate **Technical Documentation**.

For the BBK-V2 make sure the following jumper settings:

Jumper	**setting**	**description**
J4	4A 4C 4E	bus arbitration level 3
J5	1B 1E 1H	all privileges, both data types
J8	1C 2A 2C 3A 3C	32 address-bits, same as J5
J33	1B 1E	link 0 to VME-bus adaptor

- o The transputer board has two ejectors. One on the top, the other on the bottom of the front panel. Dismount both ejectors. Otherwise the board would not fit into the SUN-adaptor frame. Release the two mounting braces of the adaptor frame and insert the board into the frame. Pay special attention to firmly insert the connectors. Fix the board in the frame with the two mounting braces.

- o Insert the adaptor frame into the choosen slot. Pay special attention to firmly depress the frame into the computer. The connectors must have good contact. Use the two screws on top and bottom of the board for this aim.

- o As the final step reconnect the power cable and switch on the computer. It will perform an automatic selftest and proceed with booting itself. After some minutes you should see the **login** message. Remount the cover plates.

Now you have finished the hardware installation successfully.

Software installation

The successfull installation of the **Helios/SUN** release requires **SunOS 4.0.1** or a later version on your SUN. The kernel must have the semaphore features included. You can check the existence of this feature by the UNIX command **ipcs -s**. The total memory needed for the installation is aproximately 2 Megabyte. During the installation of the **Helios** you must have superuser permission. We strongly recommend to contact the system administrator for this software installation. If you are not truely familiar with UNIX administration avoid any experiments. It is very easy for the superuser to damage the system in a few seconds in such a way that some days of hard work are required for repairing.

First we will present you a step by step instruction to install the software. For systems not connected to a network this will fit all your requirements. Like other software installations there are more things that must be taken into account for a machine embedded into a network.

Step by step software installation instructions

o Get superuser permission. As mentioned above contact the system manager and be very carefully.

o Create a new user entry with user number **95** named **helios**. This is required only because every file is owned by somebody. All the files on the streamer tape are owned by this dummy user. This user will never **login**. If it is not possible to use **95** as user number on your system you must change the ownership of all files on the distribution tape. See the note below. The entry in the password file should look like this

```
helios:*:95:25:Helios operating system:/helios:/bin/csh
```

o Create a new user group which will have read/write permission to the VME-bus. We strongly suggest to create a group with group number **25** named **vmebus**. If it is not possible to use **25** as group number you must change the group membership of all files on the distribution tape. There is no problem, see the note after reading in the streamer tape. On a machine not connected to a network the only action is to add an entry in the file **/etc/group**. Otherwise the system manager knows how to create such a group. To create this group add the following line at the end of the file **/etc/group** :

```
vmebus:*:25:helios
```

o Give read/write-permission to this group for the VME-bus device. The VME-bus device is named **/dev/vme32d16**. First change the group membership to the **vmebus** group. Type this command:

```
chgrp  vmebus  /dev/vme32d16
```

To give read/write-permission to this device for the **vmebus**-group, type the following command:

```
chmod  g+rw  /dev/vme32d16
```

Comment:

The programs **helios, hydra, ulti** need read/write access to the VME-bus. They must reside **in** a group which has the permission to do this and the **set group id** bit on. They do not release this permission because the underlying link interface is capable to handle several open links at one time. So unfortunatly the **vmebus** permission cannot be given up.

- o The Helios software uses its own root-directory. Choose a directory for the Helios software. This depends on your disk organization and whether your system uses remote file systems via network (NFS). We suggest **/usr/helios** as the physical location. You can easily change this to your needs.

  Furthermore we strongly recommend a symbolic link in the UNIX root directory to this Helios root-directory. If you want to follow these recommendations, type this:

```
install -d -g vmebus -o helios /usr/helios
                                  # create physical
                                  # Helios root-directory
ln -s /usr/helios /helios         # create symbolic link
                                  # to this directory
```

- o Now read in the contents of the distribution streamer-tape into the newly created Helios root-directory. Let us assume **/dev/rst0** is your streamer-tape device:

```
cd /helios                        # set working  directory
                                  # to Helios root
tar xvpf /dev/rst0                # read in streamer tape
                                  # telling the actual
                                  # filenames and -sizes
```

This will take some minutes because there are about 2.0 MBytes of data on the tape.

NOTE:

If it was not possible for you to use **95** as user number and **25** as group number you can change the owner/group-number of all files read in with **tar** with the following command, assuming owner **helios** and group **vmebus**:

```
/etc/chown -R helios.vmebus /helios
```

This Helios package is intended to be used for many different transputer configurations. It it very flexible to be adapted for varying needs. In order to make the installation as easy as possible, a default configuration has been choosen. If you are using :

1. A single VMTM board at default address 0x80000000.
2. This VMTM is equipped with T800 transputers.
3. A suitable location for binary objects would be **/usr/local/bin.**
4. A suitable location for a read-only configuration file and transputer binaries is **/usr/etc**.

most of the installation is done automatically by a simple command.

If all of the four conditions listed above are true for you, you should skip the following 4 paragraphs.

- **You are not using only a VMTM board.**

  This means you are either using a single BBK-V1 or BBK-V2 board or multiple transputer boards (even multiple VMTMs). All transputer boards connected to this machine are described by a transputer-configuration file. A prototype may be found in **/helios/install/trans_conf**. This will be copied to **/usr/etc/transp/trans_conf** by a **make** command. A description of the files format, a formal specification and some examples are contained in this file. Please edit the file according to your local needs.

  If you are using a single board, you need not to edit the transputer-configuration file. There are three files with the extensions **.v1**, **.v2** and **.vmtm.** Simply copy the file corresponding to your transputer board type into **trans_conf**. All files are contained in the **/helios/install** directory. For a BBK-V2 type:

  ```
  cd /helios/install
  cp trans_conf.v2 trans_conf
  ```

NOTE:

Please make sure that **trans_conf** is a member of the vmebus group. Furthermore this file must not have write permission for other. This is a security feature checked by the link-interface. The command **ls -lg** lists you access rights, user- and group-ids.

- **You are using a T414 transputer.**

  There is a utility program **ulti** which checks and configures transputer boards. It downloads executable transputer programs. These programs differ for the different transputer types. The file-extensions **.t4** and **.t8** identify the different versions.

  Two transputer programs are supplied. The default names both have the extension **.cde**. These files will be copied by the **make install** command. Having a look at the file-sizes you can see that the default configuration have been set up for T800. If you are using T414 transputer, you should type:

  ```
  cd /helios/install
  cp trans.t4 trans.cde
  cp initc004.t4 initc004.cde
  ```

- **For binary executables /usr/local/bin is not suitable.**

  There are seven executable programs supplied with Helios now. Four of them are intented to be called by the user. See their names with a short description below:

hydra	link demon for remote transputer access
hydramon	monitor program for hydra; for use by the administrator
helios	the helios server
links	link adapter usage info program; may be called by user
ulti	the ultimate test utility; may be called by user
hinfo	remote hydra info program; may be called by user
serverwindow	the server window process; called by helios

  All executables will be copied by the **make install** command into the **BINDIR** directory, which is set by default to **/usr/local/bin**. To change this default value, you should edit the file **/helios/install/Makefile**. Near to the beginning there is a macro-definition setting **BINDIR**. You also have to edit the **serverwindow** parameter in the three files **host.con, host.local, host.remote** to reflect the correct pathname of the binary **serverwindow** object.

  Please note that you must have write permission during installation for this directory in order to copy the executables thereto. On machines using remote mounted filesystems this may be done on the remote server machine only.

- **For read-only configuration files /usr/etc/transp is not suitable.**

  Besides the location for UNIX executable programs a directory containing a configuration file and transputer binaries is needed. In order to keep transputer related files together, a subdirectory named **transp** will be created in **/usr/etc**.

  This default value is set in the file **/helios/install/Makefile**, too. The macro name is **ETCDIR** and its default value is **/usr/etc/transp**. The **ETCDIR** will be created by **install -d ...** which creates all possibly missing directories in the specified pathname. See UNIX manual pages **install(1)** for more details. All the files are used in read-only mode, only.

NOTE:

The pathname of the **trans_config** file is compiled into the programs **ulti, links, helios,** and **hydra.** You can override this definition by having an environment variable **TRANSP_CONFIG** specifying a different pathname. You can also change the definitions in the file **/helios/install/lconf.c** and recompile and reinstall the programs by means of the makefile in /helios/install.

From here on the installation proceeds for everybody.

The following commands will copy several files. During the copy some of them will be stripped and owner/group membership changed. Type the following commands:

```
cd /helios
make install
```

If you want to look at the commands which will be executed, call **make** with the **-n** flag.

- **Changing Helios configuration files**

  You have to edit the **hydra.con** ,**host.remote** and **usr/guest/host.remote** file in order to specify the name of the machine where the transputer boards are plugged in.
  The line you have to edit is:

  ```
  hydra_host = charm
  ```

  The name **charm** needs to be replaced by the **name of your machine**.
  If you dont know it, you can type the command:

  ```
  hostname
  ```

  to get the correct name. This name must correspond to an entry in the **/etc/hosts** file on every machine intended to run the Helios server. In the file **hydra.con** you can also specify the link-adaptors hydra can access initially. If you want hydra having access to all of them (you specified in the **trans_conf** file) just uncomment the line:

  ```
  #all_sites
  ```

  If not, you have to specify every link-adaptor seperatly. For the map of link numbers to physical adaptors use the **links** utility.

NOTE:

The administrator has the ability to change the access rights of a running hydra demon by means of the hydra monitor program **hydramon**. This program should only be restricted to the use of the administrator because it can disconnect remote helios server from hydra.

- **Installing the hydra demon.**

  Hydra is an internet service and must be added to the system configuration file **/etc/services** as a **tcp** service, with a unique socket number; this addition must be made on every machine which will run either **hydra** or the **server**. If you use yellow pages, the addition has to be made only there. The additional line should look like :

```
hydra                567/tcp                  #helios link demon
```

The C-shell uses advanced strategies to locate executable objects. They are based on a hashing scheme. You must type

```
rehash
```

to force the shell updating its hash tables. Otherwise it would not find **helios** though it is in your **bin directory**. This is neccessary only during installation. Afterwards there is no need to worry about (but think of having icluded the path to the helios binaries in your PATH environment variable).

- As the last steps of the installation of this software package, you should insert the following commands in a shell-script, which is automaticly executed during the system-boot. The commands initialises the link-interface and establishes a default topology on electronic configurable transputer boards, e.g. VMTM or MTM-SUN and run the hydra link demon for the remote access of the links.

```
if [-f /helios/hydra.con -a -f /usr/local/bin/hydra -a -f /usr/local/bin/ulti]; then
        (/usr/local/bin/hydra  -c  /helios/hydra.con) &
        (ulti -c)
fi
```

  Let us assume you insert this commands into the **/etc/rc.local** file. You must take care, that the **ulti** and **hydra** commands are found by that shell and that the file **/helios/hydra.con** is accessible at that time. That means the search-path includes the **BINDIR** directory mentioned above. To give the shell a chance to execute it, the according subtree of the filesystem must be mounted. The **ulti** command also accesses (for reading) all files contained in the **ETCDIR** directory.

  If everything is fine, the **ulti** command will finish silently. The **hydra** demon will print the message:

```
Hydra: running on 4 links.
```

  All neccessary configuration will be done. Additionally every transputer is checked by running a simple test-program on it. The estimated run time is a few (below 5) seconds.

Now the complete **Helios** installation has been finished. Leave the superuser status by logging out. Check your installation according to the next section.

Checking the complete installation

First you have to login as a normal user. Use your normal login sequence. Use the **ulti** utility program to make a complete check of all the transputers available in your system. First type

```
ulti -v
```

You will see a list of all the links to transputer in this machine. This list is generated from the information contained in **/usr/etc/transp/trans_conf**. Following the list is a protocoll of checking the transputer. If you have executed **ulti** with the **-c** option before, no error messages should apear. The VMTM board has been configured so that every transputer is accessable. Probably you get some errors because the VMTM has not been initialised yet. You may configure it manualy by typing

```
ulti -c -v
```

This gives you detailed information about the configuration procedure and the checking afterwards. You should not see any error messages here.

Now you may start Helios to make sure it is installed correctly. Please refer to the next paragraph.

Starting the system

This section describes how to start Helios from UNIX command level on a SUN workstation, and how to start using the system. For detailed descriptions please refer to the paper **The Helios Unix I/O Server**.

If you want to start helios on a remote SUN i.e. a SUN which is connected via ethernet to the one where the transputers are, you should not try to 'remote login' if you want to use the SunView user interface. In this case you use the help of the link demon **hydra** on the SUN with the Parsytec boards (host SUN). If you are using **'dumb terminals' (like vt100)** on your SUN you cannot use the SunView environment. Only in this case it is recommended to 'remote login' and therfore to use the **host.local** file.

- Copy the file **/helios/host.local** or **/helios/host.remote** to a **host.con** file in your **working directory**, depending on the way you want to access the transputer

  ```
  cp /helios/host.local host.con
  ```

  On a remote SUN you have to copy **host.remote**, on the host SUN you can use both, but we suggest to use **host.local.**

NOTE:

Be sure to have a symbolic link **/helios** to the physical Helios root directory on the remote Sun where you start Helios. Otherwise you have to specify the full pathname in the **helios_directory** entry in your active host.con file.

- If you want to use a fixed linkadaptor for Helios add a line

  site = <number of the link>

  in the **host.con** file. Valid numbers are **0 - 3** for a **4 user licence** and **0 - 7** for an **8 user licence**. To see the map of logical link (site) numbers and physical linkadaptors run the utility links on the host SUN. Otherwise you will get the first linkadaptor available on your hardware.

- **To start helios, you have to be in a SunView shelltool window. The server will not work correctly if you call it from a commandtool window.** Type simply

  helios

  This will start the Helios transputer operating system. First the server window should appear as a new suntools window. After the message **Booted...** comes up in this new window a second window named Console will be created after a few (up to 10) seconds. If you want more information about what is going on before the console window comes up, type **helios -o** instead of **helios** as mentioned above.

- You are asked for login. Type

  guest

  as username. Your helios home directory is now the one of the **prototype user guest**. For regular use you should establish yourself as a new user to helios. For further details about that please refer to the next section.

- Now you can enter your helios commands at the prompt in the Console window. To exit helios click with the mouse at the exit field of the server window, or enter the helios command logout in the console window.

Installing new helios users

To allow user to use their **unix** working directory as helios working directory, the system administrator should establish a symbolic link on the **unix** level in **/helios/usr** to the users directory.

```
cd  /helios/usr
ln -s  ~ <username>
```

Next he has to add an entry for every user in the passwd files in every etc directory. We suggest to change the **etc0/passwd** file and copy it to the other etc directories or make the other etc directories a symbolic link to etc0.

Then he should switch to the user account and copy the files and conf directory of the prototype user **/helios/usr/guest** to the users home directory.

```
su <username>
cp -r /helios/usr/guest/*  ~ <username>
```

At last the user needs an **appropriate host.con** file in his directory (**see section: Starting the system**) and a path variable which includes the path to the binary executables of the helios server, eg by adding the following line to the **.loginrc** file

```
setenv PATH ‘printenv PATH‘:/usr/local/bin
```

Description of the utilities links, hydramon, hinfo, and ulti

The **links** program lists all known links to transputers and their availability. It uses the link-interface described below. The transputer-configuration file **/usr/etc/transp/trans_conf** is used as the data-base for the displayed information together with the semaphore, used for inter-locking the links. Because of this the program **links** works only if it runs on the SUN where the transputer are plugged in.

Hydramon is a program to monitor the link demon **hydra**. You can display the usage of the links, **disconnect** a user from a link, **use** a link, which hydra formerly was not allowed to use or which you have given free, or give a link **free** for use by other programs.

The **hinfo** program displays informations about the usage of the links which are managed by the link demon **hydra**. It is a scaled down version of the **hydramon** program and needs the **hydra.con** file and therefore can be called on every remote SUN. Because you can only display the hydra link status, this program can be used by **everyone**.

The utility **ulti** checks the transputer and additionally configures the electronic link-switch on a VMTM board. The aim of this program is two-folded:

First it should be called from the system startup-file **/etc/rc.local** to set up a default topology on VMTM boards. (Also creating a semaphore with owner **helios**). The typical call looks like **ulti -c**.

Second it makes sure that all transputers are well and alive. It opens the link, which includes a check, whether the link-adapter is accessable on the VME-bus. After downloading a simple program (file **/usr/etc/transp/trans.cde**) to the transputer it sends a single byte through the link expecting the value incremented by one back from the transputer. The user may try **ulti -v.**

There are several command line parameters possible for **ulti**:

-v means verbose output. Usually this command is silently only errors are reported. Giving **-v** option will produce a lot of informative messages.

-c means additionally configure before checking the transputer. Default name for the configuring transputer program is **/usr/etc/transp/initc004.cde**. Using **-f** option with a full pathname changes this default.

-f followed by a full pathname changes the default value **/usr/etc/transp/initc004.cde** for the configuring transputer program into the specified file.

-t followed by a full pathname changes the default value **/usr/etc/transp/trans.cde** for the checking transputer program into the specified file.

-l followed by a link-name changes the number of links which will be handled. Usually all links are processed. If **-l** is specified only the specified link is processed.

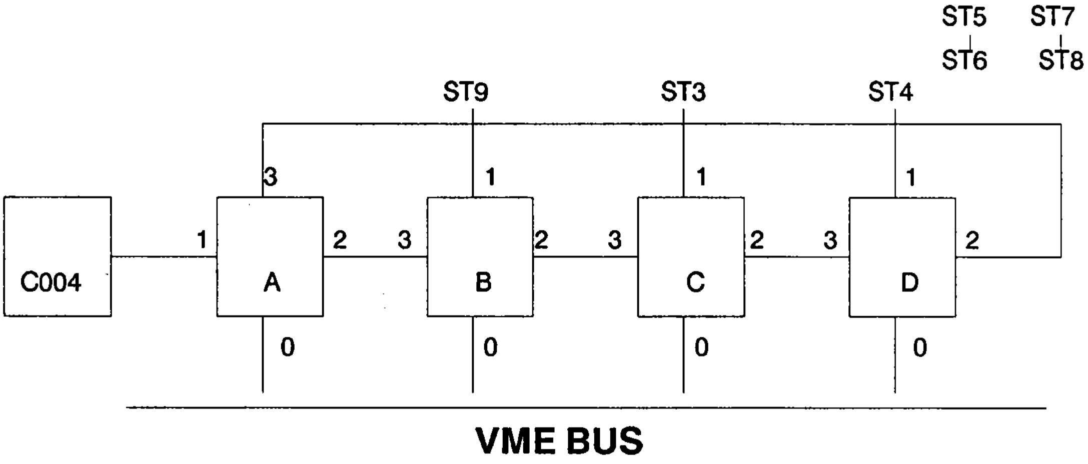
You can also reconfigure the **C004 link switch** on the VMTM board from the helios level by means of the program **prog_c004**. In order to realise your desired configuration, you have to specifiy it in the **conf_c004.c** file and to recompile the program under Helios with

```
c -o prog_c004.myconf prog_c004
```

To execute the program successfully you must run helios on the **A transputer section** of the VMTM board.

This release introduces a completely rewritten link-interface.
It is described in **/helios/install/src/link.doc**. It may handle multiple access to different links from a single program or even different programs. Furthermore it is a unique interface for all software dealing with transputers. A set of semaphores is used to prevent several users accessing the same link. This link-interface needs a description of all the installed transputer boards on this machine. A human readable configuration-file is used to describe all the installed transputer-boards. The default location for this file is **/usr/etc/transp/trans_config**. Additionally an environment variable **(TRANSP_CONFIG)** may specify a different filename to be used instead.

The configuration you get with **ulti -c** and the standard **initc004.cde** file is the following:



You can contact PARSYTEC also via **mail**. Our mailing path is:

nimra@informatik.rwth-aachen.de

Remarks and hints for further improvement are welcome. Do not hesitate to contact us in case of any trouble!

Installation Notes

Helios for the Sun Workstation

Copyright

This document was printed in the U.K.

August 1989

Preface

This note describes how to install Helios onto your Sun workstation, and gives details of the Helios Unix I/O Server. It is split into two chapters: the first describes the installation procedure, the second describes the I/O Server. It is intended that this note will supplement the standard Helios documentation, and in particular *The Helios Operating System*, which is published by Prentice-Hall.

Chapter 1

Installing Helios on Sun Workstations

Introduction

This chapter describes the Helios/Sun release and how to install it on your hardware, which must be using SunOS 4.01 or later.

Installation

The Helios/Sun product comes on a single cartridge tape, written to /dev/rst8 on a Sun3 unless otherwise requested; no special blocking factors are used. The tape contains two directories: *sunbin*, which has a number of executables to run on the Sun host; and *helios*, which contains the normal Helios files for the transputer side. The first stage in the installation process is to extract these two directories from tape, using **tar fxv /dev/rst8**; approximately 2 megabytes of disk space will be required. After extracting from tape it may be necessary to modify the owner, group, and access permission of the files to reflect the conventions at your site. Under most circumstances normal users need read access to *helios/bin*, *helios/lib*, and *helios/include*, but read/write access to *tmp* and the various *etc* directories.

The next stage is to install the various Sun executables in the *sunbin* directory, so that they can be accessed by all users. The executables exist in Sun3 and Sun4 format. In theory the hydra link daemon and the hydramon monitor program do not have to be available to ordinary users, only to the site administrator, but it may be easier during the initial stages to allow general access to these programs. The *serverwindow* and *serverwindow.sun3* programs are run by the Server as separate processes, never by the ordinary user, and should be kept in the helios directory; *serverwindow* is the Sun4 version, and *serverwindow.sun3* is for the Sun3. Finally, the I/O Server must always be accessible to ordinary users.

It will be necessary to modify the master *host.con* Server configuration file, to reflect the location of the Helios directory within the host filing system. The default Helios directory, */home/sp2sun1/user1/helios*, is unlikely to be correct for your installation. After the master copy has been updated every user should make his own copy of the configuration file in his own home directory, and always run the I/O Server from that directory, never from within the helios directory. If desirable, users could make their own copy of the helios directory and work within that, modifying their private host.con file, but this will use an additional 2 Mbytes of disk space for each user. Please note that the Server will always pick up the host.con file from the current directory, unless an alternative filename is introduced by the -C option.

The remaining part of the installation process is to determine the networking and site allocation. This depends very much on both the hardware and the users' requirements. I shall describe some of the options available and the work involved, but these should be regarded as guide lines only.

Option 1

The simplest case has a single Sun (not networked) with a single Transtech board. This configuration is intended for four users, each with just one transputer. In this case the Server will always run on the Sun with the Transtech board, so there is no need to go via the link daemon, hydra. Every user should therefore comment out the line *site = x* in their configuration file, host.con, which means that they will be allocated any free site when they run the I/O Server. They should also comment-out *run -r startns* and *waitfor /tfm* from */helios/etcx/initrc*, as there is no need to run the network server on a single transputer network.

Option 2

The next level of complexity is to have the same hardware as above, but in this case we will allow a maximum of two users, each with access to two processors. This means that site 0 uses the processors corresponding to /dev/nap0 and /dev/nap1, and site 2 uses the processors corresponding to /dev/nap2 and /dev/nap3. Under no circumstances should the Server try to use sites 1 or 3, as there is no way to detect from the Sun side that the corresponding root processors are already being used. Before running the

Server, it will be necessary to connect the processors together using the **nt_ctl** utility. Each user's configuration file, *host.con*, should contain a reference to either site 0 or site 2, but note that the Server will issue an error message if that site is already being used. As each user now has a network of two transputers, it is necessary to use the network server to boot-up the second processor, so */helios/etc/initrc* should not be modified. You should also ensure that the resource map specifies the reset driver, *tram_ra.d*, which is suitable for the Transtech scheme.

It is also possible to build networks with 3 or 4 transputers, using just the Transtech board, by adjusting the network resource maps and the host.con file as appropriate.

Option 3

The next level of complexity is to attach additional transputers to some or all of the sites, but keeping the resulting networks separate. This would allow four users, each with multiple transputers. If all four sites have the same network attached there is no reason to run on any particular site, so the host.con files should not specify a particular site. If the sites have different configurations, users may wish to run on a particular site, and they can specify this in *host.con*. Once this level of complexity is reached, it is assumed that the various users of the system will cooperate with each other.

If all four sites have different network configurations they will require different network resource maps, which is fairly straightforward as every site has its own private copy of */helios/etc*; *etc0* is used for site 0, and so on. The reset driver used for the network depends on the hardware.

Note that any sites not currently used by Helios can be used for other applications such as TDS.

Option 4

The final option allows all the sites to be accessed remotely over the ethernet. In this case it is necessary to run the link daemon, Hydra, on the Sun with the Transtech board, and this requires a number of installation steps. First, Hydra is an internet service and must be added to the system configuration file */etc/services* as a tcp service, with a unique socket number; this addition must be made on every machine which will run either Hydra or the Server. Next, the *hydra.con* configuration file must be modified, and in particular the line specifying the *hydra_host* must be changed to indicate the network address of the machine running Hydra; this network address must correspond to an entry in the */etc/hosts* file. Finally the host.con file must be modified to indicate a remote transputer box, and the *hydra_host* line must again be modified.

If desired it is possible to allow remote access to only some of the sites by listing those sites rather than *all_sites* in the configuration file, hydra.con. The remaining sites can then be accessed directly, which is more efficient than going via Hydra, but does not allow networked access. It is also possible to change which sites are accessible remotely using the hydramon program.

The Helios Unix I/O Server

Introduction

This chapter describes the various facilities provided by the Helios I/O Server, version 3.72, when running on a Unix machine. It is intended to supplement the chapter on the I/O Server in the Helios manual, and is aimed primarily at the Sun I/O Server with a Transtech transputer board (although it should also be relevant to other Unix versions). In particular, this chapter describes the various ways of configuring the I/O Server using the *host.con* configuration file. This configuration file usually resides in the current directory, and is used when the Server starts-up. It is possible to specify a different configuration file on the command line using the -C option, for example,

```
    server -C ../../dumbterm.con
```

This option is particularly useful when combined with shell aliases, allowing the user to have different commands for the different configurations.

The Windowing Interface

The Sun I/O Server can provide multiple windows on the host side, either using real windows on a Sunview display or multiple pseudo-windows with hot-key switching on a dumb terminal; the latter gives a similar environment to the I/O Server used with Helios/PC. To determine which windowing system to use the I/O Server examines the environment variable, TERM: if this is set to "sun", real windows will be used; pseudo-windows are used in other cases.

It is possible to disable multiple windows on the host side and use multiple windows on the transputer side instead, by commenting out the line Server_windows in the host.con file. In this case the transputer will run the program /helios/lib/window, which should provide multiple windows one way or another. This option is unlikely to be useful for the Sun Server.

Sunview

When running the I/O Server under Sunview, Helios will use real Sunview windows for its own windowing operations. All Helios operations which create a window will therefore create a new Sunview window. These windows use the standard Helios escape sequences for input and output, as documented in *The Helios Operating System*. In addition the I/O Server will inherit the special tty keys for *erase*, *intr*, *start*, and *stop*, and will map these onto the Helios equivalents.

Note that most programs only check the screen size at start-up, and cannot therefore be correctly used with window's that are resized during program execution. This is a result of the way in which each individual application has been written, and is not a restriction that is imposed by the I/O Server.

The I/O Server has its own window for its debugging output. This window has a control panel for the various facilities available within the I/O Server. There are buttons for rebooting Helios, for terminating the I/O Server and returning to Unix, for entering the I/O Server's low-level debugger, and for obtaining the Server status. The error logger destination can be toggled, and there is a pop-up menu for the debugging facilities. The left mouse button can be used to enable or disable all debugging options, and the right button can be used to select a particular option as follows (the keys in brackets are used with dumb terminals) :

Resources (x) List all open streams.

Reconfigure (z) Re-read the host.con configuration file. Please note that some of the options do not take effect until you reboot the transputer, and some of the options are only checked when the Server starts up.

Messages (m) Report on messages sent by/to the transputer network.

Search (s) Report all distributed searches.

Open (o) List all files being opened.

`Close`	`(c) List all files being closed.`
`Name`	`(n) Give the names of all objects Helios tries to access.`
`Read`	`(r) Report all file reads.`
`Boot`	`(b) Progress report during transputer bootstrap.`
`Keyboard`	`(k) Report all key presses.`
`Init`	`(i) Progress report while device servers are starting.`
`Write`	`(w) Report all file writes.`
`Quit`	`(q) Give a progress report while the Server is exiting.`
`Graphics`	`(g) Report any graphics operations.`

These debugging options can also be enabled on the command line. For example, **server -opr** starts the I/O Server with the *open*, *close*, and *read* debugging options enabled. This is compatible with the PC version of Helios.

For a number of reasons the I/O Server has to fork a new program whenever it creates a new window. Usually this is **serverwindow** for a Sun4, or **serverwindow.sun3** for a Sun3, but it is possible to specify some other program in the configuration file. The following example would cause the server to execute **myservwindow**.

```
serverwindow = myservwindow
```

Dumb terminals

When the I/O Server is executed from a dumb-terminal, the Server will use its own windowing system. In this environment only one window is visible at a time, although the others can be viewed by applying special key sequences. Output to non-visible windows proceeds normally, and will become visible when the user switches to that window. The termcap database and the environment variable, TERM, are used by the I/O server to interpret the standard Helios output escape sequences, and to map the terminal's input to the Helios input sequences when necessary.

The I/O Server has its own window that is not directly accessible from Helios. This window is used for the Server's error messages, and may be used as the destination for the error logger. When output is written to the Server's window, this window will pop to the foreground allowing the error messages to be observed. To disable this option, you can add the line, *Server_windows_nopop*, in the configuration file.

The I/O Server uses a number of special keys or key sequences to control reboots, debugging options, window switching, and so on. The keys for these operations may have to be different for different terminals, so they can be defined by the user by adding entries into the configuration file. A possible host.con entry is

```
    escape_sequence = k1
```

This specifies that the main hot-key is function key 1, k1 being the termcap name for that function key. Other termcap names are described in the standard Unix documentation, but the most common ones are k1-k9, which represent the first ten function keys, kh for the home key, and ku, kd, kr, and kl, for up-arrow, down-arrow, right-arrow, and left-arrow respectively.

To perform various Server operations, the user simply presses the appropriate hot-key, followed by another key, as specified below.

```
    <hot key> '1' : Switch to next window.
    <hot key> '2' : Switch to previous window.
    <hot key> '3' : Refresh current window.
    <hot key> '7' : Enter debugger.
    <hot key> '8' : Server status.
    <hot key> '9' : Server exit.
    <hot key> '0' : Reboot transputer.
    <hot key> 'a' : Toggle all debugging options.
    <hot key> 'l' : Switch error logger destination.
    <hot key> 'x' : Resource debugging.
    <hot key> 'z' : Re-read configuration file.
```

The same mechanism can be applied to the debugging options described above.

Some of the above operations are used more often than others, and it is convenient to have them as single-key operations rather than a two-key sequence. The following lines can be added to the host.con file to assign these operations to functions keys.

```
switch_forwards_key  = k2
switch_backwards_key = k3
refresh_key          = k4
debugger_key         = k5
status_key           = k6
exit_key             = k7
reboot_key           = k8
```

It is possible that your terminal has keys for which there is no termcap entry, or for which the termcap entry is incorrect. It is still possible to use these keys as escape keys by specifying the key's data in the host.con file. The key's data must always be prefixed by '#'.

```
escape_sequence = #\E^Q\012O\n
```

The previous example specifies that the main hot key generates an escape character (hex 0x1B), followed by a control-Q (hex 0x11), the octal number 12 (hex 0x0A, or ASCII linefeed), the letter 'O' (hex 0x4F), and another linefeed (hex 0x0A). Spaces can be entered by using their octal value 040, and backslash and caret can be entered by using \\ and \^ respectively.

The I/O Server will translate the termcap sequences into the Helios sequences shown below.

Termcap	**Helios**
k1-k9	Function keys 1 to 9.
k;	Function key 10.
&8	Undo.
@7	End.
kI	Insert.
kN	PageDown.
kP	PageUp.
kh	Home.
kd	Down-arrow.
ku	Up-arrow.
kr	Right-arrow.
kl	Left-arrow.
%1	Help.

If any of these are used as special keys for the I/O Server they cannot be read by Helios; any other keys are passed to Helios without translation. Helios programs should, not in general, make assumptions about the keys that will and will not be available on a particular terminal.

To perform the translation of Helios escape sequences to screen operations, the I/O Server uses the following termcap entries:

```
bl   (bell sequence; used in preference to CTRL-G)
cl   (clear screen);
cm   (cursor move);
mr and me, or so and se    (inverse/normal video);
ce   (clear to end of line);
am   (determine some of the terminal's wrapping characteristics);
ro and co   (determine the terminal size)
```

If any of the above are not defined correctly, then the Server's behaviour is undefined. In addition, the exact nature of a terminal's line wrapping may cause the display to become confused, so a screen refresh key-sequence is provided.

The Filing System Interface

The I/O Server provides two Helios servers to allow access to the host's filing system. The first is */helios*, which contains all the standard Helios files and binaries. The second is */files*, which maps onto the root of the Unix filing system, so that, */Cluster/IO/files/usr/games* is the same as the Unix directory */usr/games*. All files and directories on the Unix filing system are accessible from Helios, including networked drives. However, Helios does not gain the user any access authority other than their standard Unix ones.

The location of the main Helios directory depends on the site, and must be specified in the host.con file, together with some other files, as follows:

```
helios_directory = /home/sp2sun1/helios
bootfile         = ~/lib/nboot.i
system_image     = ~/lib/nucleus
```

The tilde characters (~) in the last two entries indicate that they are relative to the helios directory, so in most cases it is only necessary to change the *helios_directory* entry when installing Helios.

The various Helios files in the standard release occupy over a megabyte of disk space, which is compact for a complete operating system, but nevertheless it is undesirable to have more than one copy of these files on a site. This creates a number of problems. Firstly, different network maps may have to be used to boot up different transputer sites, even though the Helios initialisation file */helios/etc/initrc* only specifies one, which is usually */helios/etc/default.map*. Also, copies of Helios are serialised and there is checking within Helios to prevent multiple users from using the same copy of Helios. To overcome this, all accesses to the */helios/etc* directory and to the */helios/lib/net_serv* program are modified according to the site used. For example, if the user is connected via site 2 and tries to access */helios/etc/motd*, the server will actually access etc2/motd within the helios directory.

Because the entire Unix filing system is accessible from Helios, the user can also access the various devices and other objects. Character and Block special devices are treated as private Helios objects, and cannot be used from Helios. Symbolic links, sockets, and fifos are not supported in the current release.

The Error Logger

The I/O Server contains a device, */logger*, which may be used for error output by Helios programs. By default, all data that is sent to the logger device is diverted to the Server's own window, but an alternative destination may be specified by the user. When running under Sunview this is done by clicking a mouse button on the Logger cycle; on a dumb terminal it is done by using the key sequence, <hot key> 'l'. A Server status request will display the current logging destination.

If the logging destination is a file or both file and window, any data sent to */logger* is appended to the end of a logfile. This data may be read from Helios using standard commands, for example, cat **/logger**. If it is desired to empty the logger file, this can be done from Helios by the command-line, **rm** /logger. Alternatively, when the Server exits, any data written to the logger will be preserved in the logfile, and may be examined at leisure. The file will be cleared when the Server is run again.

There are two entries in the host.con file which control the behaviour of the error logger. The first entry, *logfile = <filename>*, specifies the file that is to be used to store logging output; the default is *logfile* in the current directory. The second entry is *logging_destination*, which can be set to *screen*, *file*, or *both*, and controls the initial logging destination. The following example entries would cause any data written to the logger to go to a file *logbook*.

```
logfile = logbook
logging_destination = file
```

Multiple Links

On many Unix-hosted transputer systems the host has multiple link adapters into a transputer network or into different transputer networks. If each link adapter is connected to a different root transputer, then it is possible to have multiple users running Helios. This combination of link adapter, root transputer, and possibly some additional transputer network is known as a site, for example a Sun with one Transtech board has four sites, allowing four users to run Helios at the same time. Any site not currently used for Helios can be used for other software, such as TDS.

The I/O Server must be able to interact with the link adapter, directly or indirectly. If the I/O Server runs on the host with the link interface, it can access the link directly. If a user wishes to access transputers in a remote machine over the Ethernet (or other local network), and still have the benefits of real windows etc., then the I/O Server must run on his own machine and interact with the link via a link daemon known as **hydra**. Communication between the I/O Server and hydra takes place using TCP/IP sockets over the ethernet.

As each transputer site may be different, there must be some way for the user to specify which site they will use. This is achieved by adding a suitable entry into host.con, and will take the form, site = 0, site = 1, etc. Note that the sites are simple integers which are mapped onto the actual hardware by the Server. On a Sun with a Transtech board, site 0 corresponds to the link device /dev/nap0, site 1 corresponds to /dev/nap1, and so on. On a Telmat SM90 with an ITFTP32 link interface, site 0 corresponds to /dev/link_0, site 1 corresponds to /dev/link_8. If no site is specified in the host.con file, the Server will choose any available site. At some future stage it is intended to provide a dialogue between the Server and the user to allow the latter to choose a site interactively.

The configuration file controls whether the Server communicates directly with a link device (in which case it has to run on the same host), or that it interacts with the link daemon, hydra. The former is significantly more efficient as it avoids the communication overhead within the Unix world, but it does not allow remote access. The relevant entry is *box* = <*hardware name*> or *box* = *remote*. If the entry specifies a particular piece of hardware, NTP1000 for the Transtech Sun board, ITFTP32 for the Telmat board, then the Server will interact directly with the link; if the entry specifies *remote*, then the Server will interact with the link daemon, hydra.

If the box is specified as remote, the Server will interact with hydra over a TCP/IP socket. This socket can be within the Unix family or within the internet family, and is controlled by the *family_name* entry in the configuration file; a setting of *AF_UNIX* or *AF_INET* should be used as appropriate. With the Unix family it is necessary to specify a name within the Unix filing system for the socket. For example,

```
family_name = AF_UNIX
socket_name = my_socket
```

This would use a Unix socket called *my_socket* in the current directory. The default family is AF_UNIX, and the default socket name is hydra.skt.

If the socket family is AF_INET, the Server will use the normal networking routines to connect to the hydra daemon. It needs to know the network name of the machine on which hydra is running, and this is specified by using the *hydra_host* entry in the configuration file.

```
hydra_host = sp2sun1
```

This host name must correspond to an entry in the */etc/hosts* file. Given the host address, the Server uses the network routines gethostbyname() and getservbyname() to obtain a socket identifier. This requires the site administrator to enter *hydra* in the file, */etc/services*, when Helios is installed. The file should be modified on each of the machines which is likely to be used for running hydra or the I/O Server. The entry should like this,

```
hydra 1234/tcp
```

where 1234 is any socket number not used by other services.

It is possible that hydra may be unable to accept a new connection immediately. This is particularly true if the system is heavily loaded by one or more users who are booting at the same time. Under these circumstances the Server will display the message, "Hydra is busy...", and retry after a short delay. The number of retries is specified in the host.con file by the entry, *connection_retries*. The default value is 5.

The protocol used between Hydra and the Server is independent of the hardware, and hence the two programs may run on completely different machines. For example, it is possible to run Hydra on a Sun4 and the I/O Server on a Sun3, or vice versa.

Hydra

The link daemon, hydra, is a separate program which is normally under the control of the site administrator, and is run automatically when the host boots up. A separate monitoring program, hydramon, can be used to interrogate hydra to determine which sites are currently in use, and by whom. In addition, it can be used to disconnect a particular Server and release the site; this disconnection occurs immediately, and may result in the loss of data, so access to hydramon is normally restricted. Hydramon also allows sites to be released, which means that hydra will no longer allow access to those sites, and allows sites to be used again.

Both hydra and the hydramon program read a configuration file *hydra.con*. A typical hydra.con file would look like this:

```
host             = SUN
box              = NTP1000
hydra_host       = sp2sun1
#family_name     = AF_UNIX
#socket_name     = my_socket
family_name      = AF_INET
connection_delay = 25
#all_sites
nap0
#nap1
nap2
#nap3
```

The *hydra.con* file closely resembles the *host.con file*, although there are fewer options. It specifies the type of host and transputer network, and the network address of the host. This network address should correspond to an entry in */etc/hosts*. Like the Server, hydra can use either Unix or internet sockets, and for internet sockets the system administrator must add hydra to the list of available services in */etc/services*, using any free socket number. Connection_delay specifies a delay in seconds between accepting new connections from Servers; the bootstrap process requires a very considerable amount of input/output, so having multiple users booting up within a short time of each other may overload hydra.

The final entries in the file specify which sites are to be used by hydra. It is possible for hydra to use all available sites, or only a selected number of sites. The latter option is useful to allow users to access particular sites without going via hydra, or even to run software other than Helios on these sites. Hydra only locks sites that are currently running Helios, so it is possible to access a site directly even if it is one of the sites accessible via Hydra.

For reliable operation it is essential that Hydra can put the link devices into non-blocking mode. Given the lack of protection in the transputer hardware, it is perfectly feasible for the transputer to crash in the middle of link traffic, in which case Hydra should be able to recover rather than hang on a read or write. If the device does not support non-blocking mode it is possible for every Helios session to hang because of a single transputer crash, and Hydra will display a warning to that effect.

Other 'host.con' Configuration Entries

There are a number of other entries in the host.con configuration file which can be changed by the user.

message_limit = 30000. This specifies the maximum size of the data vector that is used to transfer messages between Helios and the I/O Server. On a Unix system it is very expensive to have small message limits, because transferring data to and from the link involves switching between the I/O Server and the operating system. Having smaller limits can avoid problems on some other machines, but is unlikely to do so under Unix. The maximum message limit is 64000.

root_processor = /tom. Under Helios, every processor has a name. The name of the root processor is controlled by the I/O Server, and cannot change while Helios is running. The default name is /00, but it can be changed via the configuration file.

io_processor = /sun. The I/O processor behaves just like a transputer within the Helios network, and hence it too must have a network name. The default name is /IO, but this can be controlled by the user.

transputer_memory = 0x200000. On a normal system, Helios itself determines the amount of memory attached to the transputer. Problems may be experienced if the main transputer memory is followed immediately by video memory or other hardware, as Helios will attempt to use this memory as well. By specifying the amount of memory using this entry in the configuration file, Helios will not attempt to use the special areas. You should ensure that this option never specifies a larger amount than that which is actually available; if you ignore this warning, Helios will attempt to use non-existent memory. The example entry shown above specifies that the system has two megabytes of memory (using hexadecimal).

bootlink = 1. On most transputer hardware the I/O processor is connected to link 0 of the root transputer, so Helios assumes that this will always be the case. If your hardware is different you should specify an alternative by applying this entry.

To conclude this chapter, an example host.con file is now shown.

```
host          = SUN
box           = NTP1000
#site          = 0

#box = remote
#family_name = AF_UNIX
#socket_name = silly
family_name = AF_INET
hydra_host = sp2sun1
connection_retries = 10

message_limit = 60000
system_image  = ~/lib/nucleus
helios_directory     = /home/sp2sun1/user1/helios
bootfile = ~/lib/nboot.i

logfile = logbook
logging_destination = screen

#transputer_memory = 0x100000
bootlink = 1
#root_processor = /tom
#io_processor = /sun

Server_windows
#server_windows_nopop
escape_sequence =ku
switch_forwards_key = kl
switch_backwards_key = kr
#status_key = kd
#debugger_key = kl
exit_key = kd
#reboot_key = kr
#refresh_key = kd
```

With this configuration file the Server would interact directly with a link adapter rather than going via Hydra, and it would choose any free site. On a dumb terminal the cursor keys would be used for escape sequences. The other options are all standard.