The Helios PC Server

Perihelion Software Technical Report No. 1

Bart Veer

April 1989

Perihelion Software Limited
The Maltings
Charlton Road
Shepton Mallet
Somerset
BA4 5QE
England
Telephone +44 749 4203
Fax. +44 749 4977

Contents

1 Introduction

The Helios PC Server provides a complete interface between Helios running on one or more transputers and your PC running MS-DOS. This note describes some of that interface in more detail than the generic description of IO Servers provided in the Helios manual. It assumes a familiarity with both Helios and MS-DOS.

2 System Configuration

This technical report refers to version 3.60 of the PC Server and should be accurate for all 3.6x versions of this program. The PC Server is designed to run on any PC or AT or compatible, under version 3.0 or later versions of MS-DOS. It can work with any Inmos B004 transputer board or compatible, a Meiko MK026 interface board, or a Cesius Accelerator board. It provides Helios with access to the PC's filing system, screen, keyboard, mouse, serial ports, and parallel ports. It is not guaranteed to run on any other configuration, and will not provide access to any other devices. It is not guaranteed to coexist with any Terminate-and-Stay-Resident software nor with all device drivers. It is not guaranteed to run correctly on all PC or AT clones, although all reasonable care has been taken.

3 The Configuration File

The PC Server needs to know certain details of the system configuration. These are held in the host.con configuration file, which is used when the PC Server starts up. The following entries must be present in the configuration file.

```
Host = PC
Box  = B004
Message_Limit = 60000
Helios_Directory = \helios
System_Image = \helios\lib\nucleus
bootfile = \helios\lib\nboot.i
```

The Host can be PC or AT. The Box can be B004, MK026, or Cesius. The Message Limit imposes a limit on the amount of data transferred between the PC and the transputer in one message, and should be between 600 and 65000 bytes. Large values will reduce the time taken to read large files from the hard disc, but may increase delays for interactive software.

The Helios_Directory is the location within the MS-DOS filing system of the main Helios directory. Similarly the system_image is the name of the Helios nucleus, and the bootfile is the name of the nboot.i program used during bootstrap. The configuration file will also contain some device-specific entries, which will be discussed below.

4 The Link Interface

The server can use three different transputer link interfaces: Inmos B004 or compatible; Meiko MK026; and Cesius. It would need small modifications to work with other interfaces, and you should contact the board manufacturer about availability. Some aspects of the link interface can be configured in the host.con file, as follows:

```
Link_Base = 0x150
Reset_Timeout = 9999
Analyse_Timeout = 4999
```

Link_Base specifies the address of the link interface in the PC. The defaults are 0x150 for B004, 0x100 for MK026, and 0x180 for Cesius. Reset_Timeout specifies a loop count used to control resetting the root transputer. If you have any problems resetting the root you may need to increase or decrease this value. Analyse_Timeout is a similar loop count used when analysing the transputer.

5 Discs

The PC server provides access to any floppy disc, hard disc or networked disc drive, in the drive range A-Z. Under Helios these drives are accessed as /a, /b, etc. In addition there is a pseudo-drive /helios containing the main Helios directories. The PC server can locate hard discs and networked discs automatically, but MS-DOS does not provide a simple way of determining whether there are one or two floppy disc drives. The server assumes initially that there is only one floppy disc, which is drive A. If you have more than one floppy disc then you will need to add an entry in the HOST.CON file to specify which drives are floppies. This entry is of the form

```
floppies = abd
```

where the keyword *floppies* is followed by the drive letters of those drives which are floppies. The example above specifies that drives A, B and D are

floppies.

MS-DOS imposes a number of limits on file access. In particular there can be at most 17 files open at any one time, and file names must be MS-DOS file names. If more than one program writes to a file at any one time, or if one program reads from a file whilst another program writes, the result is undefined.

6 Multiple Windows

The PC Server supports multiple pseudo-windows. Each window takes up the entire screen, but the Server remembers the current state for each window and updates the screen when you switch windows. Output to windows continues whether or not the window is currently visible. Under some circumstances it may be desirable to run a window manager such as X-windows on the transputer side rather than on the I/O processor, and hence the server windows are optional. To enable server windows, you need the following line in the host.con file:

```
Server_Windows
```

The Server's window manager is a standard Helios server. Hence you can list the contents of the window directory by "ls /IO/window", and this gives you the names of the current windows. You can create your own windows using suitable system library Create() calls, or the equivalent in other libraries.

```
Object *window_manager = Locate(NULL, "/window");
Object *window = Create(window_manager, "mywindow", Type_File, 0);
Stream *stream = Open(window, NULL, O_WriteOnly);
```

Alternatively you can use the wsh command to create a new window automatically, and run a new shell in that window. Windows must be deleted explicitly, by using the rm command, for example, and do not go away just because no application is currently using them.

The Server has its own window, which cannot be accessed directly from Helios. The Server's debugging output, certain error messages, and any output sent to the Server using an IOdebug() call appear on this window. By default the Server window comes to the front automatically when any output is sent to this window. This option can be disabled by the following entry in the host.con file.

```
Server_Windows_nopop
```

However, this option should be used with care because it means that you will not notice error messages automatically.

To switch between windows there are three special key combinations. The first is Alt-F1. If you hold down either Alt key and then press function key 1, the Server will switch to the next window. When you reach the final window the Server will wrap around to its own window and then back to the first Helios window; Alt-F2 has a similar effect to Alt-F1, but moves backwards through the window list instead of forwards. The other key combination is Alt-Shift-F1. This is used to switch between the Server window and some other window. It is particularly useful when the Server has switched automatically to its own window to display some error message, because you can use it to get back to the previous window without cycling through all the others.

Each window has its own set of attributes. Hence if you disable Console Echo in one window it does not affect any others. Many applications can write to a window, but only one application should read from it any one time to avoid confusion. In particular, applications that read from the window should never be run in the background because both the application and the shell will try to read from the window and each will tend to get about half the characters.

7 Serial Ports

The PC server can provide complete control over the communications ports of your PC. To achieve this it has to work at a very low level, interacting directly with the 8250 chip, rather than going via dos or bios calls. This may cause problems on non-standard hardware, so the server only accesses the serial ports if specified in the host.con file. The following line is needed:

```
rs232_ports = 1,2
```

This causes the server to use com1 and com2 only. Note that serial ports used by devices such as serial mice cannot be accessed from Helios, because the device driver will tend to use the port at the same time as the server.

A program running under Helios may read and write from

```
/rs232/default
```

which refers to the standard communications port. This port is usually COM1; however, if you have more than one serial port (COM1 and COM2)

and you use ls to list the directory /rs232, you will discover three entries called default, com1 and com2. The name default will refer to com1, but you can rename com2 as default using the mv command, in which case default now refers to com2. Programs can, of course, refer to com1 and com2 directly. You can also change which port is the default by adding an entry to the HOST.CON file of the form

```
default_rs232 = com2
```

which makes com2 the default.

MS-DOS only defines the addresses of com1 and com2. To access com3 onwards you need to provide additional information in the host.con file. For example,

```
com3_base = 0xnnn
```

will setup support for /rs232/com3, assuming the base address of the port to be hexadecimal *nnn*. Another problem can occur if you use a serial port for a device such as a mouse, because both the server and the device driver will need to take over the rs232 interrupt vectors. By default the server takes over both rs232 interrupt vectors, but you can configure it to use only one of them by a line of the form:

```
rs232_interrupt = 1
```

For example, suppose your PC is equipped with two serial ports com1 and com2, and you want to use com1 for a serial mouse, and com2 as a Helios serial port. You would need the following two lines in the host.con file.

```
rs232_ports = 2
rs232_interrupt = 2
```

The /rs232 directory will contain a single entry /rs232/default, and this will refer to port com2.

For full details of a Helios rs232 device, please consult *The Helios Operating System* or technical note no. 6, *The Helios RS232 Device* (both available from Perihelion Software Ltd.). The PC implementation has a number of limitations. In particular, the 8250 UART chip only supports one baud rate, so the same baud rate is used for input and output.

8 Parallel Ports

The server supports access to the parallel ports from Helios, using standard BIOS calls. As with the serial ports, the file

```
/centronics/default
```

refers to the default parallel printer port, which is normally LPT1. If you have more than one parallel port then you will find entries default, lpt1 and lpt2 in the /centronics directory. Again you can use these extra entries directly, rename them using mv, or add a HOST.CON entry

```
default_centronics = lpt2
```

to change the default port when the server starts.

9 Printer

The PC server also provides a device called 'printer' which refers to either the serial or parallel ports. The directory /printers contain an entry called default, plus an entry for each serial port, plus an entry for each parallel port. You can use my to rename the required serial or parallel port as default; alternatively the entry

```
default_printer = com2
```

could be used to alter the default to the second serial line whenever the server starts.

10 X-Window Support

An implementation of X-windows on the transputer needs access to a mouse device, and a raw keyboard device. The PC Server can supply these, although they tend to be very hardware specific and hence may not work correctly on all machines. They are therefore optional, and are available only if you have the following line in the host.con file:

```
XSupport
```

The Helios /mouse device provides a Helios mouse as described in the manual. It works via a standard MS-DOS mouse device driver, usually supplied with the mouse hardware, and hence it should work on all hardware. In particular, it is known to work with PC serial mice and Microsoft bus mice, although in the case of serial mice you need to worry about clashes with the Helios /rs232 device as described above. The amount of movement needed to generate a mouse event can be configured in the host.con file, by the following line:

```
mouse_resolution = 8
```

Smaller values reduce the amount of movement needed to generate a mouse event. The Helios mouse device provides pseudo-absolute coordinates. This facilitates recovery if an event message is lost, because the lost message can be ignored completely and the mouse can still be positioned correctly. If the application needs relative mouse coordinates, the following filter routine may be used:

```
void filter_coords(int x, int y)
{ static int old_x = 0x1000000, old_y;
  int dx, dy;

  dx = x - old_x; dy = y - old_y; old_x = x; old_y = y;

  if ( abs(dx) > 0x4000 )
   { if ( abs(dx) > 0x100000 ) return;   /* initial position */
     dx = ( dx > 0 ) ? dx - 0x08000 : dx + 0x08000;
   }
  if ( abs(dy) > 0x4000 )
   dy = ( dy > 0 ) ? dy - 0x08000 : dy + 0x08000;

  feed_to_cat(dx, dy);
}
```

The raw keyboard provides Helios with key-up-key-down scancodes, and is intended for use only by X-windows. It works by taking over the keyboard interrupt vector and interacting directly with the keyboard chip, so it may not work correctly on all hardware. The keyboard device works as documented in the Helios manual, except that the data sent in the event messages are not ASCII + extensions but PC scancodes.

When the raw keyboard is enabled it is still possible to use the Server's debugging facilities, e.g. control-shift-F10 to reboot the transputer. However, there may be a slight problem with certain keyboards; when the Server gets an interrupt for key 0x12 down, it has no way of knowing whether this

key corresponds to 'w' on a QWERTY keyboard or to 'z' on a French AZERTY keyboard, for example. Hence the Server assumes that you are using a QWERTY keyboard for the purpose of analysing debugging options.

11 User-Defined Traps

It is possible to activate trap 60 under MS-DOS directly from Helios. The following example program in C would be run under Helios on the transputer. Following this we show the matching assembler program which would be need to be run on the PC under MS-DOS. Note that if the application does not send a message to the /PC stream at least once every 30 minutes, the PC Server assumes that the application has crashed and will close the stream automatically. If this happens the application should just reopen the stream.

```
/**
*** This program illustrates how to activate trap 60 on an
*** IBM PC or compatible. This allows you to install your
*** own resident modules and access them from Helios,
*** e.g. to control devices not supported by the standard
*** Helios server.
**/

#include <stdio.h>
#include <stdlib.h>               /* Needed for exit() */
#include <syslib.h>
#include <message.h>

int main(void)
{ MCB    mob;
  Object *IO;
  Stream *stream;
  Port   reply_port;
  WORD   result;

  IO = Locate(NULL, "/pc");   /* Find the server for */
  if (IO == Null(Object))     /* the host machine    */
    { printf("Unable to locate /pc - exiting.\n");
      exit(1);
    }

  /* Now open a Stream to the host */
  stream = Open(IO, "/pc", O_ReadOnly);
  if (stream == Null(Stream))
    { printf("Unable to open /pc - exiting\n");
      Close(IO);
```

```
    exit(1);
  }

reply_port = NewPort();     /* Get a reply port */
if (reply_port == NullPort)
  { printf("Unable to get reply port - exiting\n");
    Close(IO);
    exit (1);
  }

mcb.MsgHdr.DataSize = 0; /* Fill in a message structure */
mcb.MsgHdr.ContSize = 0; /* The message has no data     */
mcb.MsgHdr.Flags    = MsgHdr_Flags_preserve;
mcb.MsgHdr.Dest     = stream->Server; /* Message port for device */
mcb.MsgHdr.Reply    = reply_port;     /* Our own port           */
mcb.MsgHdr.FnRc     = 0x20765432;     /* A private function code */
mcb.Date            = Null(BYTE);     /* (see codes.h)          */
mcb.Control         = Null(WORD);
mcb.Timeout         = 10 * OneSec;

result = PutMsg(&mcb); /* Try to send a message */
if (result != 0)       /* Tidy up if it fails   */
  { printf("PutMsg returns %lx\n", result);
    Close(IO);
    Close(stream);
    FreePort(reply_port);
    return(result);
  }

mcb.MsgHdr.Dest = reply_port; /* Device's reply will be sent here */

result = GetMsg(&mcb);
printf("GetMsg returned %lx\n", result);

Close(IO); /* Tidy up and exit */
Close(stream);
FreePort(reply_port);
return(0);
}
```

```
;
; Resident program to test the Helios call-trap facility.
;
; Install a simple routine at interrupt 0x60, to be called
; by the Server when it receives a private protocol message
; for the /pc device. The routine gets a pointer to the
; Server MCB in registers ds:dx, allowing it to manipulate
; the message before it is sent back to the client - make sure
; that the data size and control size entries in the MCB are
; set correctly. The routine should return in under two seconds,
; with a 32-bit reply code in dx:ax - this is sent back as the
; message FnRc to the client. The MCB structure can be found in
; the Helios manual and in the message.h header file.
;

cseg       segment para public 'CODE'

   org     100H

   assume  cs:cseg, ds:cseg, es:cseg, ss:cseg

Init       proc near

   mov     dx,cs           ; force all segment registers to
   mov     ds,dx           ; a sensible value
   mov     es,dx
   mov     dx,offset trap  ; install routine "trap" as the
   mov     ax,2560H        ; interrupt vector for trap 0x60
   int     21H

   mov     dx,offset signon ; display a message to show that
   mov     ah,9            ; the routine is installed
   int     21H

   mov     dx,((offset Pgm_Len+15)/16)+20H ; terminate but
   mov     ax,3100H                        ; stay resident
   int     21H

Init       endp

;
; This is the routine that will be activated by the Server
;

trap       proc far

   sti                     ; make sure that interrupts are enabled

   mov     ax,cs           ; set the data segment register
```

```
   mov     ds,ax
   mov     dx,offset warn  ; display a message to show that the
   mov     ah,9            ; trap has been activated
   int     21H
   mov     dx,8765H        ; return code 0x87654321
   mov     ax,4321H

   iret

trap       endp

cr equ     0dH
lf equ     0aH

signon     db    cr,lf,'Trap 60 handler installed',cr,lf,'$'
warn       db    cr,lf,'Trap 60 activated',cr,lf,'$'

Pgm_Len    equ   $-Init ; size of this program, needed
                        ; to terminate but stay resident

cseg       ends

end        init
```

12 Debugging Facilities

The numbers of debugging flags have increased significantly since the publication of the Helios manual, and the full list is shown below. Each debugging flag can be activated on the command line, for example, by using "server -mns" to start the server with message, name and distributed search debugging enabled. Alternatively a flag can be toggled at any time using a control-shift-key combination. For example, control-shift-M will enable or disable message debugging. These debugging flags are aimed mainly at system programmers, but can prove extremely useful when developing applications. Suggestions for additional debugging flags are always welcome, with the proviso that I am running out of letters.

A	This enables or disables All the debugging options.
B	This option gives information during the Bootstrap process.
C	This option gives information about serial line Communications. Note that C is also used on the command line to specify an alternative host.con file, so this option can only be enabled at run-time.
D	Used on the command line to enter the Debugger. At run-time the control-shift-F7 option should be used.
I	This option gives information about the Servers Initialisation process, useful if the server appears to boot up the transputer correctly but nothing else happens.
K	This gives information about Keyboard events.
M	This is used for Message debugging. Details of the Helios function and error codes are held in the header file codes.h
N	This controls Name debugging.
O	This informs the user whenever Helios attempts to Open a file.
P	This informs the user whenever Helios attempts to close a file. It is not mnemonic, but C was already taken and P is next to O in the alphabet and on a QWERTY keyboard.
Q	This gives information when the Server is Quitting back to MS-DOS.
R	This gives information about Reading files.
S	This outputs a message whenever the Server receives a distributed Search.
W	This gives information about Writing to files.
X	This lists all the streams currently open. Again, it is not mnemonic but S was already taken, and it is unlikely that there will ever be a mnemonic use for X.