Inferno 3rd Edition June 2001 Revision
Release Notes and Errata

Vita Nuova
support@vitanuova.com
18 June 2001

1. Installation

If you have a previous version of Inferno installed, this one must not be installed over it. If you have a Lucent Inferno 2.3 release from 1999 or earlier, you should make a completely new installation from this CD. If you have installed the Vita Nuova 'Binary and Limited Source' CD you should also make a new installation from this CD. If you have previously installed the Vita Nuova Full Source CD from July 2000, see the section 'Updating the July 2000 release' at the end of this document. (You could also make a fresh installation in a new directory, or remove the old release and install this one if there are no files you wish to keep.)

Follow the installation instructions in ''Installation of the Inferno Software'' at the end of Volume 2. The printed copy has several mistakes:

> Page 327 notes that on Unix if the installation is done as user `root` but a user `inferno` exists the files will be owned by `inferno`. They are not. Indeed, on Unix there is no reason to install the package as the super-user `root`. Do the installation as the user, perhaps `inferno`, that is to own the files.
>
> It refers on page 327 to the Windows installation program as `Nt-386.exe`. The correct name is used on the next page in the actual instructions, namely `\install\setup.exe`.
>
> The Windows `setup.exe` will offer to create the target installation directory if it does not exist. On all other systems, you must first create the directory yourself.
>
> Solaris and some other commercial Unix systems do not read the 'Joliet' format section of the CD, which uses Unicode names, and they force the names in the non-Joliet portion on the CD to lower case, sometimes with hyphens converted to underscores. The installation script on Solaris is therefore `install/solaris_sparc.sh`. The installation script itself works despite this. Alternatively, if the CD is mounted with the option `nomaplcase` the system will not force the names to lower case and all names will appear as we intended (see the Solaris manual entry for *mount_hsfs*). Future CDs will use Rock Ridge format as well as Joliet to avoid the problem. (We attempted that this time but ran into trouble on Windows systems because of an error in the CD writing program.) Linux and FreeBSD show the names we intended.

When installing on Windows NT or Windows 2000, the installation program determines if the current user belongs to the `Administrators` group. If so, the Start Menu entry is created in the `All Users` profile, otherwise it is created in the user's private profile. On Windows 95 and Windows 98, if the user has a private profile the Start Menu entry is created there, otherwise it is created in the main system Start Menu. Furthermore, `setup.exe` looks for `Start Menu` to install the shortcut, but that name is locale-dependent, and thus the shortcut will not be installed correctly in non-English locales, although Inferno itself will be installed successfully.

When the installation completes normally, it prints ''installation complete'', but on some platforms it can then print ''Killed: ...'' followed by the command line used in the installation shell script. Provided it has said ''installation complete'' and there were no errors noted prior to that, the installation has succeeded.

The ‘‘Killed’’ message results from the installation software running inside the Inferno environment having shut down that environment. It does not mean that installation failed.

If an installation does fail, for instance by running out of space, delete everything in the target directory before retrying the installation. Also see the ‘‘Known Problems’’ section below.

2. CD Number

There is a set of six unique numbers on the back of the CD case. You should keep the case or record the numbers: they identify your subscription, and you will need them to gain access to subscriber's services we provide electronically. (They are randomly generated and usable in a 6/49 lottery!)

3. Hosted Operating System versions

The software was compiled on the operating system versions listed below. UNIX systems show the output from `uname -a` and the version of the C compiler, if known. See the section *Known problems* below for a list of known problems for any given release.

Windows Nt	Windows NT4.0 SP4
Linux	Linux vespa 2.2.9-19mdk #1 Wed May 19 19:53:00 GMT 1999 i586 ... (gcc-2.91.66)
Solaris	SunOS pazzo 5.6 Generic_105181-03 sun4u sparc SUNW,Ultra-5_10 (gcc 2.95)
FreeBSD 4.x	FreeBSD outside 4.0-RELEASE FreeBSD 4.0-RELEASE #0: ... i386 (gcc 2.95.2)
HP/UX	HP-UX hpserv1 B.10.20 A 9000/715 2013314861 ... (c89)
IRIX 5.3	IRIX invece 5.3 11091812 IP22 mips (MIPS cc)
Plan9/x86	Third Edition, updates to 5th June 2001

The Windows version has also been tested and used extensively on the following variants:

Windows ’95
Windows ’98
Windows Me

We have also installed and run the system under Windows 2000, both client and server, but there is a problem with the cursor under Windows 2000 client (see below).

Inferno source code is included for the following, but they have neither been built nor tested:

Plan 9 (mips, sparc, power)
Solaris/386
Unixware v 2.3

4. Known problems

See `www.vitanuova.com/inferno/` for current pointers to information about Inferno. You should particularly check the Frequently Asked Questions at

`www.vitanuova.com/inferno/faq.html`

and the current Bugs list at

`www.vitanuova.com/inferno/bugs.html`

The Subscriber area will include online updates after 6 July 2001.

Now, the bad news:

The Windows NT installation program will create a Start Menu shortcut that invokes *emu* with a `-g800x600` argument. The *emu* display will be incorrect if the screen width available is less than

800 pixels and is not a multiple of four. The problem can be seen if the PC is configured with a resolution of 800x600 pixels and the Microsoft Office shortcut bar is active. The simplest fix is to change the *emu* shortcut to use a `-g788x600` argument instead.

As mentioned above, the Windows NT installation program looks only for `Start Menu` not the locale-dependent name.

Changing the cursor image does not work under Windows 2000 Client; this mainly affects *acme*, which changes the cursor when rearranging or resizing frames.

The HP version of hosted Inferno was generated by HP/UX version B.10.20. On the HP platform, `emu` can currently only be run in interpreted mode; the compiled mode (ie, `-c1` option) will fault.

Some Inferno calls return error strings provided by Windows, without modification. They can be obscure: for instance, ‘‘windows error 10049’’ is produced by network calls that attempt to use symbolic names when *cs*(8) has not been started.

On all systems, the mapping from Inferno names and permissions to the underlying operating system's names and permissions needs more work, particularly on Windows. Common problems include:

File names in the Inferno environment cannot be longer than 27 bytes.

Ftpfs cannot access files with long names or names containing spaces.

The contents of `/dev/user` on Windows will contain the Windows user name, which can contain spaces.

Files created inside the Inferno hierarchy by host system applications can sometimes have odd permissions when accessed within Inferno.

Readonly files and open files cannot be removed under Windows.

In general, the mapping between Inferno users and groups and Windows/Nt users and groups is systematic, as described in *sys-stat*(2):

> *Emu* attempts to maintain a limited but consistent map between Inferno and NT worlds, specifically between Inferno names and NT security IDs. Special NT group ‘Everyone’ represents ‘other’ for file permissions. The Inferno uid is the file owner under NT; the Inferno gid reported is the first user in the file's ACL that is neither the owner nor Everyone; failing that, the gid is the file's owner.

The effects of this mapping are sometimes peculiar: for instance, something that is thought of as a user appears as the group name in *ls*, and vice-versa. Either the implementation or the mapping might need to be rethought.

bufio(2) maintains an internal list of files open for output, to support its little-used `flush`-all operation. Unfortunately, that means that if several processes use the same Bufio module instance (the result of a single `load`) concurrently for output to different files, they must separately interlock the open, create and close calls.

The file `utils/5l/thumb.c` contains comments in the C++ style, which (reasonably enough) are not accepted by the ANSI C compiler on the HP/UX system we used, and thus the current `5l` is not compiled for HP/UX; they will be changed to ANSI C comments by one of the online updates.

5. Contents

This revision offers the following, compared to the July 2000 release.

Repairs and changes

Many bug fixes and improvements appear throughout. Many commands have been revised to print usage and diagnostic consistently, and give reliable exit status for `sh`.

The shell `sh` has repairs and several visible changes: *sh-expr*(1) implements a `%` operator for remainder; a new `@` operator creates a sub-shell to execute a command, allowing the calling shell to be insu-

lated by `pctl` (see *sh-std*(1)) from changes to name space and environment; environment variables are stored in printable form; a newline is allowed after a caret; and everyone's favourite change, the colon character '`:`' is no longer reserved in argument words (allowing URLs to be given without quoting).

Inferno's Acme has been revised to match the version in Plan 9 (Third Edition). In particular, the `Edit` built-in has been added, allowing the use of structural regular expressions and the `sam` command language for efficient editing within Acme, replacing the `/acme/edit` suite of commands. The Acme panes have acquired a similar touch of colour.

Charon has also had many fixes and improvements, particularly to Javascript extraction, frames and layout code. Sometimes the 'fix' requires making Charon mimic the interpretation of incorrect HTML of other browsers. The PNG image format is now supported. Cookies are enabled by default (too many sites use them); parsing and production of the cookie file has been repaired. The progress bar is more compact. Support for longer SSL keys is enabled by default.

The software installation commands `install/*` have been extensively revised, and documented by *archfs*(4) and *create*(8). They are based (though not entirely) on Russ Cox's update package for Plan 9.

format(8) can format an ordinary file, for instance to prepare flash partition contents in a hosted environment for a native Inferno device.

`mount` allows the certificate file to be named directly.

`ns` correctly quotes the fields in its output.

`/services/server/config` no longer gives the unauthenticated ('`nobody`') option to *styx*(8); some unused entries have also been deleted.

srv(8) passes all arguments to servers it spawns, including the command name. It also ensures each server has its own process group, file descriptors and name space.

stack(1) has a new `-p` option to add names to the source file search list.

deflate(2) correctly detects end-of-file; *inflate*(2) correctly decodes the combined compressed code-length tables

Tk's handling of objects in canvases is better: raise and lower work properly; stipple is implemented.

Tk now knows that a window pops up when resized, and adjusts the z-order accordingly.

Tk's scroller always expands the fraction of a `moveto` when it evaluates the scrolling command.

New commands and modules

ftpfs(4) provides a way to make a remote FTP site visible in the Inferno name space (it was documented in the printed manual but not previously included with the system).

listen(1) provides a convenient way to listen for incoming calls to one or more Inferno services, with optional use of *ssl*(3) for authentication and encryption. This can replace the clumsy use of *srv*(8).

dial (see *listen*(1)) is the complement of *listen*; it dials a service, with optional authentication and encryption.

lockfs(4) enforces multiple reader, exclusive writer access to the contents of a name space.

The device *prof*(3) serves a name space for controlling the profiling of Limbo modules, and retrieving the resulting data. The module *profile*(2) offers a convenient interface, relating the profiling data to the source code. Finally, *prof*(1) is the command line interface to enable profiling and display the results.

`uuencode` and `uudecode` interpret a format used to encode binary data printably on Usenet and in mail messages; see *uuencode*(1).

Read(1) writes to standard output the result of a single *sys-read*(2) of a given number of bytes from standard input, with optional seek offset. (See *getlines* in *sh-std*(1) for a way to read a line from standard input.)

Tcs(1) uses *convcs*(2) to offer character set conversion on files.

wm-misc(1) mentions `wm/mand`, a browser for fractals, and `wm/polyhedra`, a polyhedra viewer

cfg(2) provides a module to read configuration files.

dividers(2) provides user-draggable dividing lines to separate Tk widgets, allowing screen space to be allocated to widgets by dragging a dividing line.

imagefile(2) offers support for reading PNG image files

Interface changes and extensions

`Url` (`/module/url.m`) now requires an `init` function to be invoked before other functions in the module

convcs(2) has changed its interface significantly (see the manual page); it also offers support for UTF-7. Even the module name has changed, to `Convcs` (from `ConvCS`).

plumber(8) now returns an error to a message's sender if it cannot be plumbed, as was previously documented; it handles `^` correctly in regular expressions

readdir(2) returns all file names in union directories

string(2)'s quoting and unquoting functions are correct and more efficient

styxlib(2) forces an internal process into a new empty name space to allow detection of the last unmount of the served space by a file server

translate(2) supports writing Unicode characters in hexadecimal using Limbo's `\u`*XXXX* syntax

workdir(2) returns a better guess at the current directory on native Inferno (hosted implementation will be supported in the next update)

cs(8) keeps a cache of recent translations; *ipsrv*(8) uses UDP/IP by default, not TCP/IP, for domain name lookup

httpd(8) can now be said to work as documented

wm-sh(1) provides a new control file `/chan/shctl` to allow it to be kept informed of the current directory (and `/lib/wmsetup` defines a Shell function to do that) and to implement buttons below the title bar, as with *mash-tk*(1).

A collection of playing card images has been added in `/icons/cards`.

`/lib/wmsetup` defines a `cd` function to keep `wm/sh` informed of the current directory; the `Web` menu item is now called `Charon`; a new `Manual` entry invokes *wm-man*(1); and the game `Tetris` has been added to the `Misc` submenu.

The manual pages *draw-font*(2), *draw-image*(2), and *draw-screen*(2) now document `display` and `screen` members of various data structures.

security-auth(2) and *security-login*(2) note that `keyring.m` must be included.

In *sh*(2), the function `exec` has been renamed `run`, to reflect more accurately what it does.

In *sys-pctl*(2), the type of the second parameter to `Sys->pctl` is actually `list of int` not `list of ref Sys->FD`.

ssl(3) documents the new `encalg` and `hashalg` files, which list the supported algorithms.

canvas(9) describes the new `-winding` option

`/services/cs/services` lists the Software Download Server port and removes others that are obsolete.

`/services/server/config` no longer gives the unauthenticated ('`nobody`') option to *styx*(8); some unused entries have also been deleted.

Limbo compiler

The Limbo compiler correctly clears reference values (eg, `list`, `ref` and `array`) when they go out of scope when necessary (eg, in loops), causing the storage to be reclaimed, and the values to be correctly `nil` when the scope is next entered.

A bug that caused temporaries sometimes to be reused too early has been fixed.

The compile-time evaluation of floating-point > does not calculate >= instead.

A little context has been added to 'syntax error' messages.

Compilers and architectures

The system now supports the Thumb variant of the ARM architecture, including full interworking of Thumb code and 32-bit ARM code; there is a Thumb JIT compiler for the virtual machine, and support for Thumb mode in the disassembler and debugger. The Thumb compiler is `tc`; the existing ARM linker `5l` links both ARM and Thumb code.

A bug in `5l` has been fixed that could very occasionally place a literal pool inside a case table.

The compilers find include files correctly under Nt.

Hosted and Native Inferno

Emu has the following fixes and improvements:

support for logging of memory pool usage

FreeBSD support uses *rfork* not *pthreads*.

Linux support no longer relies on being able to set the TSS register.

A memory leak when a kernel process exits has been fixed.

A start has been made on tidying up the audio support; the `svp.c` file has gone and platforms that do not currently support audio no longer must include a stub driver.

The Linux and FreeBSD ports use an alternative implementation of Inferno graphics under X11 that should allow the program to run under 16-bit graphics and higher.

For the native kernels only:

The native kernel implements Rob Pike's 'lexical names' (as in Plan 9), which helps provide more sensible data to *workdir*(2). The hosted Inferno implementation of lexical names will appear in a future update. Note that `/os/port/ns.c` no longer exists, and native kernel configuration files should no longer mention `ns` in the `port` section. Drivers that serve a directory hierarchy might need to support the `DEVDOTDOT` value for the table index in their `devgen` function.

`os/ip/bootp.c` will correctly time out if there is no response. `os/ip/ihbootp.c` will now work with an RFC1084 BOOTP server. Note that both are likely to be replaced in future by Limbo applications that do BOOTP and DHCP.

`os/pc/cga.c` provides a replacement for `screen.c` for use when only a simple CGA console is needed or possible.

The file descriptor array is correctly freed when the file descriptor group is closed, fixing a memory leak.

A few files that were left off the July 2000 CD have been included this time.

ARM/Thumb kernels have been implemented; those implementations are not included in this update because the hardware is specialised but ask if you could use them.

The remaining points are common to both hosted and native Inferno:

A write on closed pipe does not produce an exception if the process has been killed.

Devbwrite will not lose memory if an error is raised by the device driver (eg, interrupt).

ssl(3) has `encalg` and `hashalg` files

Kfs(3) offers `readonly` and `readwrite` commands, and an `ro` (readonly) option to the `init` request.

Srv(3) rejects attempts to create an unusable name containing a '/'.

The `NODEVS` flag of *sys-pctl*(2) is correctly copied when the name space is duplicated.

SA1100 native

A new `archether` function in `arch*.c` (when required) is invoked by `devether.c` to discover the configuration of Ethernet devices.

`devuart.c` supports the use of XON/XOFF; the first line of the `stat` file gives the current UART settings; it no longer panics if an overrun occurs; and `setlength` accesses the correct structures.

The CS8900 driver `ether8900.c` is more general, supporting the I/O port interface as well as the memory interface.

Obsolete definitions have been removed from several include files (eg, `io.h`) and obsolete (typically #ifdef'd code) has been removed from source files.

6. IDEA"

The software includes an implementation of the IDEA encryption algorithm, for non-commercial use. IDEA was patented by Ascom-Tech AG (European patent EP 0 482 154 B1, US patent number US005214703, and patent filed in Japan). At time of writing, there was no licence fee required for non-commercial use. If you intend to use IDEA encryption commercially with Inferno, you should consult `http://www.it-sec.com/idea_lic_policy.html` for the current licensing policy of iT_SEC Systec Ltd, which currently holds the patents and trademark. Note that IDEA is not required for successful use of Inferno. It is not by default enabled by *rstyxd*(8), and otherwise is used only by SSL3 (for the browser), in `/appl/lib/crypt/ssl3.b`, where it can be disabled.

7. Updating the July 2000 Full Source release

A set of update packages for the July 2000 release is provided in the directory `/updates` on this June 2001 CD. You should install the updates as the host operating system user who owns the Inferno files and directories on your system. You might like to take a backup copy of the existing tree, just in case. Do the following to update the installation. (Next time we hope to provide a more automated scheme. You might check `www.vitanuova.com/inferno/` to see if there are any more recent instructions.)

1. Copy the directory `updates` and its subdirectories from the CD to a directory `updates` in the root of your existing Inferno hierarchy. Make sure there is sufficient space in the file system holding that hierarchy. If all platform files are copied and installed, about 45 Mbytes will be needed to hold the compressed update packages in `updates`, with a further 35 Mbytes needed for an temporary uncompressed copy of the largest package, and allow 10 Mbytes for new additional material, giving 90 Mbytes in all. The `updates` directory can be removed after installation.

2. Start the existing Inferno *emu*; it will be quicker if you use the `-c1` option to force compiled mode. It is best to use only the Inferno console; do not start the window system, since the updates will change files in the running system. On the other hand, it is a good idea to make the window in the host operating system a scrolling one, so that you can scroll back to see any errors. All following commands are run in the Inferno environment.

3. Change to the directory `updates` and load the standard Inferno shell module:

```
cd /updates
load std
```

4. Unpack the updated installation software:

```
gunzip <install.tgz | {cd /; gettar}
```

5. Update the installed Inferno, source and utility source directories using the script `applybase`:

```
sh ./applybase
```

That script updates the `inferno`, `src` and `utils` packages. You will see one warning:

```
skipping /dis/install/inst.dis: locally modified
```

because that file was updated by step 4. You might see other warnings if you have modified any other files from the original release.

6. Update one or more platform specific files for your platform(s). The directories are named after the platforms: `Solaris` for Solaris, `Plan9` for Plan 9, `Nt` for all Windows systems, and so on. For each *platform* that you run, do:

```
for (a in platform/*) {install/inst -v -t $a}
```

7. Quit *emu*. The new version of *emu* will be called `emu.new` in the platform-specific directory (eg, `Solaris/sparc/bin/emu.new`). Rename the old `emu` file as `emu.old`, then rename the new `emu.new` as `emu` on Plan 9 and Unix systems, or `emu.exe` on Windows. When run, it should announce itself as ‘‘Inferno Third Edition (18 Jun 2001)’’.