Subject: Fugue.3 version of Interlisp-D now available
To: LispUsers^.pa
From: LispSupport.pa
Reply-to: LispSupport.pa

This message announces the availability of the Fugue.3 release of Interlisp-D -- the third official release of Fugue. Fugue was first released as Fugue.0 on 30 June 1983, and Fugue.2 was released on 18 August 1983. Fugue.1 was not released. Significant changes have taken place in Fugue.3; this note lists those changes, together with a summary of the important changes from Fugue.2 and Fugue.0. Each item is more fully described later in this message.

Please send all comments, suggestions, and bug reports to LispSupport.pa. (Please don’t use LispUsers^.pa.) Your input is much appreciated.

The following items are new or revised in this release:

```
<3.1> Release files: Note new location
      LispUsers files: some moved to <Lisp>Library>
      New or greatly improved packages (TEdit, Lafite, Maintain, among
others)
      Documentation files
<3.2> Window/display system changes
      DISPLAYSTREAM datatype eliminated (merged with STREAM). Many programs
        which manipulate DISPLAYSTREAMs must be recompiled, a few may need
        to be edited.
      Various improvements to Window system.
<3.3> 1108 (Dandelion) specific improvements
      Improvements to reliability and performance
      Additional floppy support: can format, write
<3.4> New interface for printing
      PRINTERMODE is gone
      FONTPROFILE now has separate Interpress entry
<3.5> Improvements to network protocol implementations
      NS protocols: a number of bugs fixed, still some performance problems
      File operations to PUP FTP-only hosts now supported
      Echo server, user, tracing for Ethernet testing:
<3.6> Miscellaneous new functions: APROPOS, PLAYTUNE, MACHINETYPE,
SETPASSWORD
<3.7> Known problems with this release
```

The following items were also part of the Fugue.2 release:

```
<2.1> Multi-process architecture extended
<2.2> TeleRaid: a remote debugging aid
<2.3> Timeouts for SAVEVM, RECLAIM in background now notice mouse action.
<2.4> EDITEMBEDTOKEN is now "&"; affects both DEDIT and EDITF. (formerly
<esc> or *)
<2.5> PROMPT#FLG now initialized to T: history event numbers are printed.
<2.6> Miscellaneous new system variables and function.
```

The following items were also part of the Fugue.0 release:

```
<0.1> Fugue runs compatibly on all three machines of the 1100 family
<0.2> Window System changes and additions
<0.3> DEDIT, TTYIN in standard loadup
<0.4> All Alto disk partitions of 1100s and 1132s accessible
<0.5> New implementation of Leaf
<0.6> Incompatible changes in low-level Ethernet code
<0.7> DURATION, PROMPTFORWORD packages now in standard loadup
<0.8> Additional logical arithmetic functions (inspired by CommonLisp)
<0.9> Access to system window and background menus
```

```
==================================================================================
<3.1> RELEASE FILES: NOTE NEW LOCATION
```

By popular demand, we have reverted to the practice of keeping the "current" release on

```
      [Phylum]<Lisp>Current>
```

rather than on <Lisp>Fugue>. Users at sites other than Palo Alto should wait for an announcement of the availability of files on their local file server from their system administrator.

The following files are all available on <Lisp>Current>:

 * Lisp memory images files:

```
      Lisp.SYSOUT            standard release virtual memory image
      Small.SYSOUT            smaller version without Interpress, or NS
protocols.
```

For the user’s convenience, we also provide two memory images with the current versions of various <Lisp>Library> and <LispUsers> packages already loaded:

```
      Full.SYSOUT            Lisp.SYSOUT, plus current versions of:
                               TEDIT, FILEBROWSER, LAFITE, READSYS
      Demo.SYSOUT            Full.SYSOUT, plus many graphics demos
```

Since the <Lisp>Library> packages on which these SYSOUTs depend (particularly TEdit and LAFITE) are under rapid development, these two latter memory images will be more frequently updated.

```
 * BCPL initialization for 1100 (Dolphin) and 1132 (Dorado)
      [unchanged since Fugue.2]:
      Lisp.run

 * Microcode for 1100 (Dolphin) [unchanged since Fugue.2]:
      DolphinLispMC.eb      for 3MB Ethernet
      XMBDolphinLispMC.eb      for 10 MB Ethernet
      X3DolphinLispMC.eb      both 3 and 10MB Ethernet, but no floating
point

 * Microcode for 1132 (Dorado) [unchanged since Fugue.2]:
      DoradoLispMC.eb

 * Initial microcode for 1108:
      LispSAx000Initial.db

 * Initialization, command files:
      Init.CIS, Init.KSA        should be copied to {DSK}INIT.LISP;
                        give this file name if prompted
                        for the name of an init file. These files
                        used to be on [phylum]<Lisp>
      NewUserDisk.<site>
      UpdateLisp.<site>        for <site> = CIS, KSA
      InstallLisp.othello        othello command file for installing
Lisp.Sysout
      Install<sysout>.othello    for sysout = Lisp, Full, Demo
```

* Sierra/Mesa 10.0 tools can be used for installing Interlisp-D on Dandelion from inside Tajo:

```
      InstallMu.bcd            InstallMu lisp _
[phylum]<Lisp>Current>Lisp.sysout
      LispTool.bcd            Window-based version of above
```

LISPUSERS FILES: SOME MOVED TO <LISP>LIBRARY>

Formerly, the <LispUsers> directory contained not only packages submitted by users, but also some packages which were maintained by LispSupport. In order to promote <LispUsers> and to regularize the release cycle, we have moved the files listed below to <Lisp>Library>. (Note that among other changes, the new INIT.<site> initialization files for {DSK}INIT.LISP have new values for LISPUSERSDIRECTORIES and DIRECTORIES). See {Phylum}<Lisp>Library>LispLibrary.TTY for summary.

New (not previously announced):

ABC, EXPORTS.ALL (access to internal system data structures)
CMLSPECIALFORMS (from Common Lisp: CATCH and THROW, LET and LET*, LIST*, and DEFUN)
DLIONFS (experimental version of disk driver for Dandelion)
FILEBANGER (test network/file system software)
FTPSERVER (single-user Lisp PUPFTP server)
GATEWAY (forward packets between 10MB and 3MB ethernet)
LISPLIBRARY.TTY (mini index/documentation; similar to LISPUSERS.TTY)
MAINTAIN (access to modifying Grapevine distribution lists)
MAKEDIR (construct the pup-network naming database for use by GATEWAY)
PUPIDSERVER (tell 10MB Ethernet hosts their PUP number)
RDSYS, READSYS, REMOTEVMEM, VMEM (support for "TeleRaid" remote debugging)
UPCSTATS (microcode-PC sample statistics for Dorados)

New versions of files from <LispUsers> (older versions were there for Fugue.2):

BIG (change font sizes for demos)
CMLARRAY (Common Lisp array facililty: multi-dimensional, "open-coding" etc.)
CIRCLPRINT (circular structure display)
COLOR, LLCOLOR, HLCOLOR, COLORDEMO, COLORPOLYGONS (support for color)
COLORPOLYGONS (color demo)
EVALSERVER (EVAL on other machines across net)
FILEBROWSER (file display, like Neptune)
GRAPEVINE, MAILCLIENT, MTP (low level support for PUP mail protocols)
ICONW (transparent icon windows)
LAFITE (mail program)
LOGOCLOCK (clock in the Interlisp-D logo window)
PRINTER (support for a low-cost dot-matrix printer)
MENUEDWINDOW (attach menu’s to windows)
POLYGONS (graphics demo)
READAIS (read .AIS sampled graphic images files, transform, halftone)
RS232, RS232CHAT, RS232FTP, RS232EXEC, RS232LOGIN
(for RS232 serial port: drivers, dial-out "chat", MODEM protocol FTP, dial-in exec)
SINGLEFILEINDEX (indexed listing with table of contents)
TEDIT, TEDITFIND, TEDITHCPY, TEXTOFD (Interlisp-D Text Editor)

Files moved directly from <LispUsers> (unchanged since Fugue.2):

BITMAPFNS (bitmap utilities)
CHECKSET (make sure files are compiled)
CLISPENG (program in English)
COMPILEBANG (compile without asking questions)
COMPILEFORMSLIST (compile variable-form into single function)
COMPMODEREC (implementation-dependent record declarations)
COREHAX (who’s swapped in)
DATABASEFNS (Masterscope aid)
DECL (type declarations)
GCHAX (looking at the GC tables)
HANOI (graphics demo)
LAMBDATRAN (new kinds of functions)
LOGOCLOCK (analog clock in Interlisp-D logo window)
LOADDM (load Alto-style .DM files)
PEANO (graphics demo)
PQUOTE (print QUOTE as ’)
READNUMBER (menu-based number pad)
SAMEDIR (MAKEFILE checks that output is on same directory as old version)
WINK (Marilyn Monroe winking)

The following <LispUsers> packages remain on <LispUsers>, although some may move to Lisp>Library> in the future. Contributions welcome!

New (since Fugue.2):

ANIMATE (Bobrow.PA: smooth dragging bitmap; big arrow & hand)
BLACKOUT (HThompson.PA: black screen with bouncing cursor like DMT)
DISPLAYSTATS (Levy.PA: stats package interface to Grapher, to show dynamic call graph)
EDITHIST (DDyer@ISIB: keep track of edit history for files)
EMACS (Roach.PA: very small subset of EMACS built on top of TEDIT)
EXECFNS (Shulman@Rutgers: logical names for search paths in file names)
FACEINVADER (Shulman@Rutgers: game - shoot bouncing face)
READACFONTFILE (Lim.PA: Read .AC font files into internal format)

New versions (edited or recompiled since Fugue.2)

ACTIVEREGIONS (Barstow@SDR: mouse-sensitive regions)
BLTDEMO (Denber.wbst: bouncing ring & box)
EDITBITMAP (AHenderson.PA: More bitmap manipulations a la Notepad)
EDITMACROS (JonL.PA: useful tty-editor macros)
GRAPHER (VanLehn.PA, Kaplan.PA: lay out graph structure on screen)
HISTMENU (Bobrow.PA: access history list via menu)
IMAGEDEMO (Burton.PA: halftone graphics demo)
INMENU (Stefik.PA: different pop-up menus)
MAKEGRAPH (AHenderson.PA: built on Grapher, build graphs interactively)
NQUEENS (Schoen@SDR: Graphics demo of N-Queens problem)
ODOMETER (Sybalsky.PA: sequence of small windows roll like auto mileage gaugue)
PACMAN (Denber.Wbst: Good imitation of the arcade game)
PROFILETOOL (Yonke.PA: window/inspector for variables)
PROMPTREMINDERS (JonL.PA: periodically remind of events)
RPC, LUPINE, REMEVALSERVER, REMEVALUSER (HThompson.PA: Cedar remote procedure protocol)
SOLITAIRE (Sheil.PA: graphics demo)
STYLESHEET (Stansbury.PA: create block of menus with multiple selections)
TTY (Stefik.PA: simple set of interactive TTY I/O functions)
TTYIO (Reid Smith@SDR: simple TTY I/O functions)

Old (unchanged since Fugue.2)
CROCK (Roach.PA: analog clock face)
DRAWFILE (Masinter.PA: read in Alto-Draw format files)
GLISP, GEV, GEVDEMO (Novak@UTexas-20: object-oriented data-independent programming)
HEADLINE (AHenderson.PA: make a big title on screen)
LANDPRESS (Halasz.PA: print press sideways)
LOADFILES (Kaplan.PA: load in group of files)
MATHFNS (Roach.PA: math fns, e.g., trig, log, hyperbolic fns & complex arithmetic)
MOUSECONFIRM (Yonke.PA: ask for click of mouse)
MULTIMENU (Schoen@SDR: create menus associated with main window, according to context)
MULTIW (Lane@SUMEX-AIM: hierarchical windows, which move, close, etc. in groups)
NOTEPAD (AHenderson.PA, Burton.PA: Artwork editor, brushes, symmetry)
PAGEFULLCTRL (Stefik.PA: different scrolling mode)
PAGEHOLD (JonL.PA: another variation on scrolling mode)
PATCHUP (Bobrow.PA: make patch file)
PERFORMTRAN (Kaplan.PA: new "perform" keyword)
SAMPLER (Denber.wbst: graphics demo from Upstate New York)
SIGNAL (HThompson.PA: Cedar/mesa signal mechanism)
SYSTAT (Roach.PA: window with monitor of system activity)
TMENU (Bobrow.PA: unread menus, windowshades)
TRACEIN (DonC@ISIF: single-step, trace, debug)
TRAJECTORY-FOLLOWER (AHenderson.PA: cause snake to crawl across a trajectory)

DOCUMENTATION FILES

A major revision of the Interlisp Reference Manual will be available by the end of this month. An announcement about ordering information will be sent to this distribution list. Until the manual is printed, it will be made available on {Phylum}<LispManual> as a series of Press files. Of probable interest are the following chapters:

```
      ChapID.press (Interlisp-D specifics, files, process world)
      ChapWP.press (window package)
      ChapDT.press (display tools: DEdit, TEdit, TTYIN, etc.)
      ChapEther.press (Ethernet information)
```

In addition to some new versions of the documentation files on <Lisp>Library> and <LispUsers>, there are two documentation files on [Phylum]<Lisp>Current>:

```
      Fugue3Release.TXT, .press (this message)
      GettingStarted.doc, .press (revised version: How to run Interlisp-D
at Xerox)
```

As usual, please send all problem reports, questions, to LispSupport.PA. We want to hear from you!

<3.2> WINDOW/DISPLAY SYSTEM CHANGES

DISPLAYSTREAM datatype eliminated (merged with STREAM)

In the move toward fully device-independent graphics, the DISPLAYSTREAM datatype has been eliminated; it is completely subsumed by the STREAM datatype. DSPCREATE returns a STREAM, and all the DSP--- functions work on this kind of object, without any coercion. DISPLAYSTREAMP is true for STREAMs returned by DSPCREATE. FONTCREATE has been changed so that it coerces the device attribute of a given font descriptor if possible instead of causing the "wrong font device" error. INTERPRESS is now a recognized font device, e.g., in DEFAULTFONT and FONTCREATE.

***** Many programs which manipulate DISPLAYSTREAMs must be recompiled or, in some cases, edited. *****

The new system is source-compatible with the previous system, with the exception of code that makes direct reference to the DISPLAYSTREAM record, either by field accesses or by (type? DISLAYSTREAM --) constructions. Direct references to fields of the DISPLAYSTREAM record should be changed to use the functional interfaces; (type? DISPLAYSTREAM X) should be replaced with (DISPLAYSTREAMP X). Other code does not have to be rewritten, but may have to be recompiled. Specifically, any function that calls (DSPxxx NIL ...) for xxx= XPOSITION, YPOSITION, XOFFSET, YOFFSET, LINEFEED, TEXTURE will have to be recompiled, because these forms compiled open as direct field accesses in the old system. The safest thing to do is to BCOMPL any file that manipulates with the display. If the compilation succeeds, then the resulting .DCOM file is compatible with both old and new systems. If the compilation fails because the DISPLAYSTREAM record no longer exists, you must edit the file. Many Lispusers packages have been recompiled for this reason.

Miscelaneous improvements to the window system:

The scroll handler now blocks so that systems activity no longer stops when the cursor is in a scroll bar.

Moving windows from off the screen back onto the screen causes the newly exposed parts to be redisplayed.

PAINTW now brings up its menu in the background as well as in the window being painted. Thus if the painting window should be closed accidentally, it is still possible to quit PAINTW easily.

If a window is created without a title, a subsequently added title will be added on top of the window, not within it. In other words, the new windowheight will be the old windowheight plus the titlebarheight. Anything displayed in the window between its creation and its titling will not be obscured by the titlebar.

==

<3.3> 1108 (DANDELION) SPECIFIC IMPROVEMENTS

IMPROVEMENTS TO RELIABILITY AND PERFORMANCE

A number of microcode and low-level system problems which have caused numerous crashes have been found and fixed in the recent past. Thanks to all of the initial users who took the time to report problems and let us TeleRaid them. While some problems still remain, the system is far more stable, and is faster thanks to new CONS and garbage-collection microcode.

ADDITIONAL FLOPPY SUPPORT

Fugue.3 contains a preliminary version of support for the Dandelion floppy disk. You can now write to and read from the floppy disk drive on 1108s. The floppy drive device is called {FLOPPY}. Lisp uses a format compatible with Mesa so that Lisp can (usually) read floppies written by Mesa 10.0. There are 2276 pages available for the user’s files on each Pilot floppy. For each file stored, there are two pages of storage overhead.

To format a floppy disk, insert it in the drive, and call (FLOPPY.FORMAT Name) where Name is a string or atom which will become the name of the disk. Formatting will erase the old contents of the disk. (Note that formatting the floppy uses enough cycles of the IO-Processor to make the system drop some keystrokes typed while formatting.)

To write to or read from a floppy disk, use FileBrowser, DIRECTORY, LOAD, OPENFILE, READ, PRINT, BIN, BOUT, GETFILEINFO, SETFILEINFO, GETFILEPTR, SETFILEPTR, etc. I.e., all standard I/O operations work.

There are still some problems with the floppy driver: e.g., incorrect error messages; no compaction of free blocks (so that creating and deleting files may result in a situation where the floppy device thinks there is no room even though very little space is in use); no monitor locks (avoid simultaneous access from separate processes!); and no interrupt protection (interrupting writing the floppy with a control-D may leave it in a state in which no new files can be written.) Use for backup, file storage and transfer seems to be reliable.

==

<3.4> New interface for printing

PRINTERMODE is gone: it is now inferred from the printer name.

The system now automatically selects the print format and transmission protocol for a printer; it is no longer necessary to set a global (PRINTERMODE). The variable DEFAULTPRINTINGHOST can be either a single printer name or a list of names. Names can either be the names of PUP-based press hosts (e.g., QUAKE, MENLO) or of NS-based Interpress hosts (e.g., LISPPRINT:AAAI, PAPERMATE:PARC).

The system uses the function (PRINTERTYPE host) to determine the type of a printer. If host is an atom, PRINTERTYPE first looks for a PRINTERTYPE property on host. If it doesn’t find one, names with a colon in them (e.g., LISPPRINT:) are considered to be INTERPRESS printers, and otherwise, the type is assumed to be the value of DEFAULTPRINTERTYPE, initially PRESS. If host is a list, e.g. (INTERPRESS GUTENBERG), then CAR is taken as the PRINTERTYPE.

The functions EMPRESS, PRINTERSTATUS, PRINTERPROPERTIES are now all "generic": they call the appropriate functions for formatting, transmission and interrogation according to the type of the printer.

When DEFAULTPRINTINGHOST is a list of printer names, the first "appropriate" printer is used; for example, if you EMPRESS or LISTFILES an Interpress document, it will first look for a printer which can print Interpress documents, while a Press file will go directly to a press printer.

It is possible to add new printer types which key directly into LISTFILES and the HARDCOPY option in the background menu. For example, the PRINTER package on <Lisp>Library> defines a new kind of printer type for printing on a local low-cost dot-matrix printer. Printer types are registered in the list PRINTERTYPES, which is a list of the form (printertypes (property value) (property value) ...). Each kind of printer (Press=Dover/Penguin, Fullpress=Raven, Interpress=8044/5700) registers what kinds of files it can print directly, how to convert to a known format, how to scale bitmaps, create files from bitmaps, and print bitmaps. The variable PRINTFILETYPES documents the known kinds of files.

{LPT} DEVICE NOW RECOGNIZES PRINTER NAMES AS NAME FIELD

Now {LPT}FOO, where FOO the name of a printer, will print on printer FOO when closed. For example,

```
      (SETQ FILEHANDLE (OPENSTREAM '{LPT}QUAKE 'OUTPUT]
      (PRINT "This will be printed." FILEHANDLE]
      (CLOSEF FILEHANDLE]
```

will cause a file containing the sentence "This will be printed." to be printed on the printer QUAKE.

FONTPROFILE NOW HAS SEPARATE INTERPRESS ENTRY

The fontprofile package now has, in addition to entries for default DISPLAY and PRESS fonts, a separate entry for INTERPRESS. The default FONTPROFILE for INTERPRESS has been modified to reference MODERN and CLASSIC fonts rather than TIMESROMAN and HELVETICA.

===

<3.5> Improvements to network protocol implementations

NS protocols

Fugue.3 contains a preliminary version of support for NS filing to Xerox 8030 file servers. The functions DIRECTORY, FILEBROWSER, INFILE, COPYFILE, LOAD, and MAKEFILE with NS file servers. Any device with a colon in its name is presumed to be accessible with NS protocols rather than PUP, (e.g., {StarFile:}.) The general format of NS fileserver device names is {ServerName:Domain:Organization}; the device specification for an 8000-series product may contain the domain and organization; if not supplied directly, then they are obtained from the defaults, which themselves are found by a search for the nearest ClearingHouse. (Users may set CH.DEFAULT.DOMAIN and CH.DEFAULT.ORGANIZATION to be a default domain and organization instead. CH.NET.HINT is a hint to the clearinghouse-search path if your machine is not directly connected to a network with a clearinghouse.)

A number of bugs have been fixed in the implementation of NS Filing protocols since Fugue.2. We are still having intermittent performance problems, however; we believe that some combination of circumstances is causing Lisp to confuse the file server. File transfers often work, but sometimes connections time-out or bog down in the middle of transfer.

File operations to FTP-only PUP hosts now more fully supported

The implementation of the PupFtp protocol (which supports transferring files sequentially only) has been improved. When sequential access to a file is appropriate, the use of PupFtp generally results in considerable speed improvement over Leaf, particularly for writing files on an IFS. Interlisp-D now tries to use PupFtp where possible for SYSOUT and for the destination file of a COPYFILE. Users can indicate that a file is going to be accessed only sequentially by including the attribute/value pair SEQUENTIAL T in the list of MACHINE.DEPENDENT.PARAMETERS passed to OPENFILE; then PupFtp will be used, if possible. The system now automatically shifts to the FTP protocol where Leaf is not available. If for some reason your file server supports PupFtp but you do not wish to use it, you can set the internal variable \FTPAVAILABLE to NIL. (Note that PupFtp has a much shorter timeout fuse than Leaf; thus files which are processed sequentially, but very slowly, should use the Leaf protocol if possible.)

FTP also supports an enumeration protocol; functions DIRECTORY, FILDIR, the FILEBROWSER Lisp Library package, now all work for any remote hosts which support enumerate in PupFtp.

Echo server, user, tracing for Ethernet testing:

The ethernet drivers have both NS and PUP echo-servers built into them, which will respond to the appropriate echo-me packets from other machines on the net. (E.g., it is possible to echo from a NS server to a Lisp workstation). In addition, Lisp has echo-users, documented in the new reference manual Ethernet chapters.

==

<3.6> Miscellaneous new functions:

(APROPOS <pattern> <allflg> <quiteflg>) Scans all symbols in the system for one which has <pattern> as a substring, and prints out a line for each relevant item of that atom; relevant items are function definitions, global bindings, and non-null property lists. If <allflg> is NIL, then atoms without any relevant items are omitted and also "internal" atoms are omitted ("internal" currently means "pname begins with a \" or "result of GENSYM"). If <quietflg> is non-null, then no printing occurs, but a list of all selected atoms is returned.

(PLAYTUNE <Frequency/Duration.pairlist) will map down its argument, expecting each entry to be a dotted pair of frequency and duration and will beep the 1108 keyboard buzzer at that frequency for that amount of time. A NIL for the frequency means to turn the beeper off for that amount of time. Makes no sound on an 1100 or 1132. Note that the default "simulate" entry for control-G (ASCII BEL) on the 1108 will use PLAYTUNE to cause a short beep.

(MACHINETYPE) -- returns DORADO (for 1132), DOLPHIN (for 1100), or DANDELION (for 1108).

(SETPASSWORD host user password directory) -- sets values in the internal password database as if typed in via LOGIN.

==

<3.7> KNOWN PROBLEMS WITH THIS RELEASE:

* If you LOGOUT while some other process is actively writing a file, it is not guaranteed that anything reasonable happens to the file. The process system allows processes to prevent LOGOUT before completion (by using a BEFOREEXIT DON'T process property), but not all system facilities have been modified to use this mechanism.

* The {DSK} device on 1108s is only "ephemeral" and files placed there will not survive a re-loading or re-booting. The DLIONFS package on <Lisp>Library> is an experimental version of a local disk management system, but you must explicitly load it in.

* The caret is still global, rather than being a part of the window structure. This means that there are times where a caret gets left up in a window when it should have come down.

Users may find that a tty window (much like a break window) pops up unexpectedly for a process that apparently had no need for it. We have removed most such cases from system code; however, there are several cases in which user code can cause this behavior, where in earlier releases the code went unnoticed. A typical such case is where user code calls one of the DSPxxx operations, defaulting the displaystream. Unless the code REALLY wants to access the primary output of the process, the code should explicitly specify a destination.

The swapping file (Lisp.VirtualMem on 1100s and 1132s) must still be a "small" number of disk segments. Swapping performance will degrade drastically when this number of segments is exceeded, and RAID will inform you of such happening. Generally, the setup of a disk partition with a swapping file of about 10000 pages is adequate for most users, but some have needed to go to 12000; "setup" by CREATEFILE.RUN on a newly-installed disk will keep the number of segments as small as possible (often just one or two), and a few extensions beyond that will probably not cause any harm.

Most system packages were written before multi-processing was added to Interlisp-D; some of these packages rely to some extent on there being only one process running, and access and alter global state accordingly. We have corrected many of these conflicts but have not yet completed this task. Avoid running more than one instance (from different processes) of DEdit, EMPRESS (hardcopy), MAKEFILE, the compiler, or Masterscope.

===

The following changes were announced with the Fugue.2 release:

===

<2.1> Multi-process architecture extended

Affects "primary" I/O, (TTYDISPLAYSTREAM),
Window "switching" between processes; new HELP interrupt.
It is important for ALL USERS to read the documentation relating to this.

Although the multi-processing environment is largely upward compatible with Chorus, there are some important differences in the behavior of mouse functions and menus.

* Global input/output state is now per-process

There are several entities in the Lisp system that are "global" in Interlisp-10, and used to be global in Interlisp-D. These now have separate bindings in each process: primary input and output, terminal input and output, primary read table and terminal table, and the dribble file. This change is largely transparent to user code; however, any code containing the form (TTYDISPLAYSTREAM) MUST be recompiled, as this form had a macro that explicitly accessed a variable that is no longer global.

The default initial values for primary and terminal i/o when a process is created are "place holders". When the process actually tries to print/read to the primary input/output or the T stream without having installed some real stream as primary (using INPUT, OUTPUT or TTYDISPLAYSTREAM), a TTY window is created for the process. The user-settable variable DEFAULTTTYREGION is the region where these default windows come up. The function (HASTTYWINDOWP process) is true if the process has a tty window other than the initial "place holder"; (HASTTYWINDOWP NIL) refers to the current process.

Break windows are positioned near the TTY window of the broken process, as determined by the variable BREAKREGIONSPEC. This variable is a region whose LEFT and BOTTOM are an offset from the LEFT and BOTTOM of the TTY window where the break window should go; its WIDTH and HEIGHT determine the size of the break window.

* Keyboard input & tty switching.

The model for specifying which process gets keyboard input is that "to type to a process, you click in its window." Any process desiring keyboard input should put its process handle as the PROCESS property of its window(s) [using WINDOWPROP]. The function TTYDISPLAYSTREAM does this automatically when the ttydisplaystream is switched to a new window. The window manager notices when you click in a window that has a PROCESS property, and makes that process be the TtyProcess, which causes all keystrokes to go to that process. Whenever the TtyProcess is switched, typeahead is saved, so that the input goes to the process that it was (presumably) intended for.

This default mechanism can be overridden. Writers of code that need more careful control over when and how the TtyProcess is switched, in particular those who called PROCESS.SWITCH.TO.WINDOW or used the PROCESS.IDLEFN, PROCESS.ENTRYFN and PROCESS.EXITFN properties in the old Process implementation, should read the relevant portion of the Process documentation for details of the new mechanism.

Programs that used to switch to PROMPTWINDOW to read are now discouraged from doing so, to avoid contention for that window. The function PROCESS.READ still exists, but you may not need it anymore.   A process can simply READ, and the reading will happen in the process's own tty window, popped up if need be. An attempt to read underneath the mouse automatically grabs the tty process.

(READP T) now returns NIL immediately if the caller is not the TtyProcess (and there is no saved input). Implication: programs that called READP (or its internal subfunction \SYSBUFP) and then flashed a caret or took other action to indicate their being alive and/or receptive to input should consider testing (TTY.PROCESSP) first. The function (WAIT.FOR.TTY) efficiently blocks until (TTY.PROCESSP) is true.

* Interrupt characters.

Some interrupts, including BREAK (^B), are still fielded by the TtyProcess. However, ERROR (^E) and RESET (^D) are preferentially taken by the mouse process if it is not "idle" (otherwise by the tty process, as before). The HELP interrupt (^H, though many users move it to ^G) puts up a menu of processes asking which one the user desires to interrupt, and then performs the interrupt in that process.

* Miscellaneous

The mouse now runs in its own process, called MOUSE, separate from the other background tasks, such as writing dirty pages back to the disk.

The default value of the RESTARTABLE argument for ADD.PROCESS is now to never restart. If you want a process to restart when a HARDRESET occurs (the Fugue.0 default), specify a RESTARTABLE of HARDRESET.

Scroll holding works for the tty window of each process. Of course, if you use the normal mechanism, and the hold happens in other than the tty process, you have to click in the held window before you get to type something to release the hold.

There is a new item in the system background menu labeled PSW, which brings up a process status window.

==

<2.2> TeleRaid: a remote debugging aid

The 1108 has no Raid debugger (since Raid is written in BCPL), but does provide a low-level "TeleRaid" service when it would otherwise fall into RAID. When the machine is in TeleRaid, it can be debugged from another machine, or in some cases the user can issue commands to continue or break into the computation in limited ways. 1100s and 1132s can also run

TeleRaid.

Once a machine is in TeleRaid, then another may be used to debug it by calling (TELERAID <host>), where <host> is the Pup name or number of the machine running the TeleRaid server. The function TELERAID is not in the standard loadup (but is in the FULL.SYSOUT loadup); to load it, you must load the files READSYS.DCOM, RDSYS.DCOM, VMEM.DCOM, and REMOTEVMEM.DCOM from <Lisp>Library>. The user interface for TELERAID is similar to that of Raid; type ? for an enumeration of commands.

When the maintenance panel of an 1108 indicates a Lisp-induced halt, pressing the UNDO key starts the TeleRaid server. TeleRaid changes the cursor to the TeleRaid cursor, takes over the ethernet controller and keyboard, then waits for commands from either the Ether or the keyboard. Keyboard commands are given by holding down the control key (or PROPS on the standard Dandelion keyboard) and striking another key. As you strike a key, the cursor changes to give feedback about the command you are about to execute; the command is executed when you let go of the key, but if you let go of the command key first, the command is aborted. The commands are as follows:

```
    ^N - like Raid ^N: continue the computation.
    ^D - like Raid ^D: do a HARDRESET.
    ^B - cause a break underneath the error.
    ^P - display the machine's Pup host number (decimal) in the maintenance
panel
             (for remote debugging).
```

TeleRaid refuses to execute the "^B" command if the user is executing in a place (e.g., inside of a garbage collection) that would be dangerous to allow a user break.

You can put an 1100 or 1132 (Dorado or Dolphin) into TeleRaid server mode by giving the "R" command to Raid and thus remotely debug it or make use of TeleRaid's ^B command (nothing corresponding to this command is available in standard RAID).

==

<2.3> Timeouts for SAVEVM, RECLAIM in background now notice mouse action.

The background RECLAIM and SAVEVM are no longer handled by PERIODICALLYRECLAIM on PROMPTCHARFORMS, but run in the background process, and now pay attention to mouse actions as well as typein. A background RECLAIM happens if the user has made no keyboard or mouse action for RECLAIMWAIT seconds. A background SAVEVM happens, if the vmem conditions are right, if the user is in tty input wait and has made no keyboard or mouse action for SAVEVMWAIT seconds (note: this is a change in the meaning of SAVEVMWAIT).

==

<2.4> EDITEMBEDTOKEN is now "&", affects both DEDIT and EDITF. (formerly <esc> or *)

The variable EDITEMBEDTOKEN holds the token used in the embedding commands of both the display and tty-based editors. Its initial value is &, which is a change for BOTH editors (formerly, it was <esc> for DEDIT, and * for the Teletype-style). Users are, of course, free to change the value in their init files.

==

<2.5> PROMPT#FLG now initialized to T: history event numbers are printed.

The variable PROMPT#FLG now initialzied to T, so that history event numbers are printed out at the executive, making it easier to use history commands. Formerly, PROMPT#FLG was initialized to NIL.

==

<2.6> Miscellaneous new system variables and function:

MAX.SMALLP, MIN.SMALLP, MAX.FIXP, MIN.FIXP, NETWORKOSTYPES;
(RESTART.ETHER)

MIN.SMALLP, MAX.SMALLP, MIN.FIXP and MAX.FIXP: [Variables]
These are respectively the min and max on the range of the functions SMALLP and FIXP.

NETWORKOSTYPES   [Variable]
For proper service from file servers other than the Xerox IFS, you should add entries to the system a-list NETWORKOSTYPES associating the host name (all uppercase) with its operating system type, currently one of TENEX, TOPS20, UNIX, VMS. E.g., (ADDTOVAR NETWORKOSTYPES (MAXC2 . TENEX)).

(RESTART.ETHER)   [Function]
Under rare conditions, the Dandelion Ethernet driver can get into a bad state where the Ethernet appears totally unresponsive. Calling RESTART.ETHER will try to reinitialize all the appropriate items to get you going again.

==

The following items were also part of the Fugue.0 release

==

<0.1> Fugue runs compatibly on all three machines of the 1100 family: The 1100 (Dolphin), the 1108 (Dandelion) and the 1132 (Dorado)

Fugue Interlisp-D will run on the 1108 (Dandelion processor) as well as the 1100 (Dolphin) and 1132 (Dorado). 1108 configurations supported include the 8011 (10MB SA1000 disk), 8012 (29MB SA4000 disk), 8014 (40MB Quantum disk) with memory sizes from 1 MByte to 1.5 MByte. All versions are completely compatible -- it is possible to take ANY Interlisp-D memory image and move it directly from one machine to another. The 1108 Interlisp-D is implemented solely in microcode and Lisp; it has no Mesa nor BCPL, and its microcode fits in the standard 4K control store. It can be installed from floppy disks or from a file server. Any 8010 workstation with sufficient memory and a Pilot Logical Volume with approximately 12000 pages can run Interlisp-D, without disturbing Mesa or Star.

==

<0.2> Window System changes and additions:
     Incompatible change -- (CURRENTDISPLAYSTREAM) removed
     New functions -- OPENWP, OPENWINDOWS, CREATEREGION,
       BITMAPWIDTH, BITMAPHEIGHT, BITSPERPIXEL,
       PRESSBITMAP, FILLCIRCLE, DEFAULTFONT

****incompatible change****

The function CURRENTDISPLAYSTREAM, and the notion of a "primary" display stream have been removed from the system. Display functions now default their destination stream to the "primary output", just as other output functions do. If a DSPxxx function is called with displaystream NIL, and the primary output is NOT a stream of type display, an error is generated. Any user code that refers explicitly to CURRENTDISPLAYSTREAM must be changed; in addition, user code that calls displaystream operations defaulting the displaystream argument may need to be examined.

Calls to CURRENTDISPLAYSTREAM with the intent of determining the font of a prototypical display stream can use the newly provided function DEFAULTFONT, below.

New functions:

(DEFAULTFONT DEVICE FONT NOERRORFLG)   [Function]

Returns the font that is used as the default for the device DEVICE. If FONT is a font, it replaces the default for the device DEVICE.

The functions OPENWP and OPENWINDOWS are now synonyms of (and the preferred names for) ACTIVEWP and ACTIVEWINDOWS, respectively.

(CREATEREGION LEFT BOTTOM WIDTH HEIGHT)   [Function]
  Returns an instance of the REGION record, with LEFT, BOTTOM, WIDTH and HEIGHT components.

Functions BITMAPWIDTH, BITMAPHEIGHT, and BITSPERPIXEL were added to facilitate access to the fields of a bitmap. The function PRESSBITMAP is now in the standard loadup.

(FILLCIRCLE X Y RADIUS TEXTURE DISPLAYSTREAM)   [Function]
  Fills a circular area of radius RADIUS about the point (X, Y), in the window or displaystream DISPLAYSTREAM with the texture TEXTURE.

==
<0.3> DEDIT, TTYIN in standard loadup

DEDIT:

DEDIT is installed as the "standard" editor. DEDIT documentation is in the Interlisp-D Users Guide and will be in the new Interlisp Reference Manual. The version in Fugue is a completely new re-implementation of the version of former releases, and should be quite apparently faster. The function EDITMODE now takes TELETYPE instead of STANDARD to describe the "TTY" editor.

New Features:

A Break command (does a BREAKIN AROUND the current expression). DP now allows a second argument which it will interpret as a property to be edited, i.e. DP(FOO MACRO). The single argument form gets the whole property list as before.
Shift selection (COPY) remains active even when DEdit is not. Thus one can unread expressions from DEdit windows into the typescript window. (Selections are otherwise disabled when they would be inappropriate, e.g. under nested computations).
(RESETDEDIT) completely reinitializes DEdit if it gets badly scrambled.
Switching to tty (or other) editors: Subcommands on Edit allow multiple options of using DEDIT, TTYIN, or the (old) teletype editor on either the current expression or on its definition.

One can now give TTY editor commands directly into the typein buffer by prefixing them with ^Z. This gives full access to the programmable characteristics of the TTY editor -- you can define your own edit macros or use the more complex ones (SWAPC, IFY, BQUOTE). A read macro on ^Z interprets the rest of the line as an EditCom which will be executed when the line is closed. Likewise, you may enjoy ^S (Substitute old new) and ^F (Find) commands. Commands are defined by editing the variable DEDITTYPEINCOMS in what should be a fairly obvious way, and then RESETDEDITing.

TTYIN:

TTYIN is installed as the standard "line buffer" -- the mini-editor for keyboard input from the so-called T stream. TTYIN features editing of the type-in buffer with mouse or control-key commands, and "automatic word completion".

==
<0.4> All disk partitions of 1100s and 1132s are now accessible from Interlisp-D

The low level implementation of the disk drivers has been substantially revised, with a consequence that Interlisp-D can access partitions other than the one which was booted. If the other partition is password-protected, Lisp insists on the correct password before accessing any files. Partitions are denoted by {DSK1} for Partition 1, {DSK2} for Partition 2, etc. All file operations (including SYSOUT, the filebrowser, DIR command, etc.) all work for other partitions. However, it is not currently possible to run a sysout stored on another partition.

==

<0.5> New implementation of Leaf

Fugue contains a new implementation of the Leaf filing protocol, which tolerates simultaneous access to the same server from different processes and permits more overlapping of Lisp computation with file server operations, allowing for improved performance. A corollary of the latter that you should be aware of is that a file is not closed the instant that CLOSEF returns--Lisp is closing the file "in the background". It is therefore more important than ever that you exit Lisp "normally", via (LOGOUT), or (LOGOUT T), rather than boot the machine or exit via Raid.

==

<0.6> Incompatible changes in low-level Ethernet code:
PUP record declarations, GETPUP, etc.

The ethernet code was substantially rewritten for Fugue to accommodate the 10MB and Network System protocols, and also to put into Lisp all the background code formerly written in BCPL. A number of incompatible changes resulted: the PUP record changed, neccessitating at least the recompilation of all pup user code; GETPUP returns a brand new pup, rather than copying the incoming pup into a user-supplied pup datum; SENDPUP synchronizes differently with the transmitter; CREATESOCKET and FLUSHSOCKET were renamed OPENPUPSOCKET and CLOSEPUPSOCKET.

When a machine starts up, the Ethernet code gets the time from a server on the local net, if possible. The information supplied by the server includes the local time zone. Users who are not on an Ethernet and not in the Pacific timezone need to set the global variable \TimeZoneComp in the local disk’s INIT.LISP. The value of \TimeZoneComp is the number of hours west of Greenwich (negative if east).

==

<0.7> DURATION, PROMPTFORWORD now in standard loadup

From DURATION: the functions SETUPTIMER and TIMEREXPIRED? are to help write code that "times out"; there are several new I.S.OPRS to facilitate scoping such code. Complete documentation is in the new manual.
From PROMPTFORWORD: a new function PROMPTFORWORD has replaced the LispUsers package called READAWORD. It provides a system-level alternative to ASKUSER, with prompting similar to what the Alto Executive provided for login and password reading. Complete documentations is in the new manual, and for a short time will continue on <Lisp>Fugue>Promptforword.press.

==

<0.8> Additional logical arithmetic functions (inspired by CommonLisp)
LOGNOT, BITTEST, BITCLEAR, MASK.1’S, MASK.0’S,
LOADBYTE, LDB, DEPOSITBYTE, DPB, ROT

The following "logical" arithmetic functions are derived from Common Lisp, and have both macro and function definitions (the macros are for speed in running of compiled code). They have the following interpretations:

```
  (LOGNOT x)             == (LOGXOR x -1)
  (BITTEST x mask)       == (NOT (ZEROP (LOGAND x mask)))
  (BITCLEAR x mask)      == (LOGAND x (LOGNOT mask))
  (BITSET x mask)        == (LOGOR x mask)
  (MASK.1’S pos size)    == (LLSH (SUB1 (EXPT 2 size) pos)
  (MASK.0’s pos size)    == (LOGNOT (MASK.1’S pos size))
  (LOADBYTE x pos size) == (LOGAND (LRSH x pos) (MASK.1’S 0 size))
  (DEPOSITBYTE x pos size byte)
                         == (LOGOR (BITCLEAR x (MASK.1’S pos size))
                                   (LLSH (LOGAND byte (MASK.1’S 0 size))
pos))
```

The notion of a "byte specifier" has been added; from a pair of positon and size arguments, a byte-spec is constructed by the macro BYTE [note reversal of arguments as compared with above functions]

```
  (BYTE size pos)
```

Similarly, the macros BYTESIZE and BYTEPOSITION will select out the two fields. [currently, byte-specs are implemented as a typerecord; Common Lisp leaves unspecified whether there are range limitations on "size" or "pos"]

Two more "byte" functions are provided, with compiler macro support also:

```
  (LDB bspec val)     == (LOADBYTE val (BYTEPOSITION bspec) (BYTESIZE bspec))
  (DPB n bspec val)   == (DEPOSITBYTE val (BYTEPOSITION bspec)
                                      (BYTESIZE bspec)
                                      n)
```

The final function in this series is not quite so easy to describe -- ROT for "Rotate bits in field".

```
  (ROT x n fieldsize)
```

is a slight extension of the CommonLisp ROT function.   It performs a bitwise left-rotation of the integer x, by n places, within a field of fieldsize bits wide; bits being shifted out of the position selected by

```
  (EXPT 2 (SUB1 fieldsize))
```

will flow into the "units" position. The optional argument fieldsize defaults to "cell" size (the integerlength of the current maximum fixp), and must either be a positive integer, or else be one of the litatoms CELL or WORD. In the latter two cases the appropriate numerical values are respectively substituted.
A macro optimizes the case where n is 1 and the fieldsize is 16 (or WORD).

==

<0.9> Access to system window and background menus:
     New functions: DOWINDOWCOM, DOBACKGROUNDCOM
     New vars: xxxMenu and xxxMenuCommands for xxx in {Window, Icon, Background}

There are six user-accessible variables for controlling the contents of the standard Window menu, Icon menu and Background menu. For each menu, there is a variable xxxMenu that caches the actual menu, and a variable xxxMenuCommands that holds the commands for the menu, in the format of an ITEMS field of a MENU. The menu is recreated from the list of commands whenever the xxxMenu menu variable is set to a non-MENU (e.g., to NIL). The variables are:

WindowMenu and WindowMenuCommands -- the menu (and its commands) that is brought up in response to a right button in a window. A command in WindowMenuCommands should return a function name that will be APPLYed to the window.

IconWindowMenu and IconWindowMenuCommands -- analogous, used in response to a right button in a shrunken window.

BackgroundMenu and BackgroundMenuCommands -- the menu (and its commands) that is brought up in response to a right button in the background. A command in BackgroundMenuCommands should return a form that will be EVALed.

For example, to add commands to the window menu, simply add the commands to WindowMenuCommands, and set WindowMenu to NIL. To change the font used in the menus, change MENUFONT and set all the xxxMenu variables to NIL.

The following functions provide a functional interface to the interactive window operations so that user programs can call them directly:

(DOWINDOWCOM WINDOW)   [Function]

If WINDOW is NIL, it calls DOBACKGROUNDCOM.
If WINDOW is a shrunken window, it brings up the IconWindowMenu.
If WINDOW is a unshrunken window, it brings up the WindowMenu.
The initial items in these menus are described above. If the user selects one of the items from the provided menu, that item is APPLYed to WINDOW.

(DOBACKGROUNDCOM) [Function]

Brings up the BackgroundMenu. The initial items in this menu are described above. If the user selects one of the items from the menu, that item is EVAL’d.