Subject: Fugue.2 Release of Interlisp-D
To: LispUsers^.pa
From: LispSupport.pa
Reply-to: LispSupport.pa

This message announces the availability of the Fugue.2 release of Interlisp-D -- the second official release of Fugue (a preliminary version was released on June 30, 1983). Significant changes have taken place since since June 30; this note lists those changes, together with a summary of the important changes from the original Fugue release.

Synopsis of changes follows; first 10 items are the newer things since the first Fugue release. Each numbered item is more fully described later in this message.

[1] Multiple 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.
[2] Xerox Network Systems protocols for filing supported, in addition to printing
[3] Dandelion's floppy disk drive -- Interlisp can read from it
[4] TeleRaid: a remote debugging aid in lieu of Raid.
[5] Timeouts for SAVEVM, RECLAIM in background now notice mouse action.
[6] EDITEMBEDTOKEN is now "&", affects both DEDIT and EDITF.
  [Formerly, it was "*" for Teletype-style editor, and <esc> for DEDIT.]
[7] PROMPT#FLG now initialized to T: history event numbers are printed.
[8] Miscellaneous new variables:
  NETWORKOSTYPES, MAX.SMALLP, MIN.SMALLP, MAX.FIXP, MIN.FIXP
[9] RESTART.ETHER -- function to re-initialize ether drivers.
[10] New capabilities for RS232 users:
  Functions for control of modem lines, RS232READSTRING for "fast" input
  Asynchronous input buffering at slow speeds
  "Roll" mode for RS232CHAT window, a {RS232} device interface

[11] Fugue runs compatibly on all three machines of the 1100 family:
  The 1100 (Dolphin), the 1108 (Dandelion) and the 1132 (Dorado)
  An operators' guide for the Dandelion is available
[12] Window System changes and additions:
  Incompatible change -- (CURRENTDISPLAYSTREAM) removed
  New functions -- OPENWP, OPENWINDOWS, CREATEREGION,
   BITMAPWIDTH, BITMAPHEIGHT, BITSPERPIXEL,
   PRESSBITMAP, FILLCIRCLE, DEFAULTFONT
[13] DEDIT is in standard loadup
[14] TTYIN installed as default linebuffer editor
[15] All Alto disk partitions of 1100s (and 1132s) are now accessible from Lisp
[16] PupFtp protocol now supported; DIRECTORY works on remote file servers
  Leaf protocol re-implemented with more flexibility
[17] Incompatible changes in low-level Ethernet code:
  PUP record declarations, GETPUP etc.
[18] System-level LispUsers packages now in standard loadup
  DURATION (functions SETUPTIMER and TIMEREXPIRED? and new I.S.OPRS)
  PROMPTFORWORD (formerly called READAWORD)
[19] Additional logical arithmetic functions (inspired by CommonLisp)
  LOGNOT, BITTEST, BITCLEAR, MASK.1'S, MASK.0'S,
  LOADBYTE, LDB, DEPOSITBYTE, DPB, ROT
[20] Access to system window and background menus:
  New functions: DOWINDOWCOM, DOBACKGROUNDCOM
  New vars: xxxMenu and xxxMenuCommands for xxx in {Window, Icon, Background}
[21] New and revised LispUsers packages; some incompatible between Chorus and Fugue
[22] Finally, some items that have NOT been fixed in this release

To run in this system, obtain NEW VERSIONS of the following files from [Phylum]<Lisp>Fugue> for your local disk:

for 1100: Lisp.run, INIT.LISP, DolphinLispMC.eb, AltoD0MC.eb
(use XMBDolphinLispMC.eb for 1100's with 10MB Ethernet)
for 1132: Lisp.run, INIT.LISP, DoradoLispMC.eb, AltoD1MC.eb
for 1108: none (get installation instructions from file
[Phylum]<Lisp>Fugue>1108OperatorsGuide.press/tty

For all machines, there are new versions of the canonical site-dependent init files (to be stored as {DSK}INIT.LISP on every partition/machine) on [Phylum]<Lisp> (note, no sub-directory). We have created versions for the CIS, CSL, KSA areas at PARC -- site liasons at other locations should modify these to fit the local environment (e.g., to indicate where the default printer, file server, user init files etc. are stored). In addition, [Phylum]<Lisp> contains updated versions of the NewUserDisk.<site> and UpdateLisp.<site> files.

The "standard" release sysout is [Phylum]<Lisp>Fugue>Lisp.sysout. In addition, there is a smaller loadup available as SMALL.sysout that omits many of the new networking code and is about 800 pages smaller. Furthermore, there is an enhanced sysout on [Phylum]<Lisp>Fugue>FULL.sysout, which includes TELERAID and the LispUsers packages TEDIT (text editor), LAFITE (Xerox Grapevine electronic mail service), and FILEBROWSER (menu-driven interface to file directories),. These packages are under rapid development and have not been included in the standard release at this time.

Users at sites other than Palo Alto should wait for an announcement of the availability of files on their local file server from their local Lisp liason.

We have been hard at work completing a major revision of the Interlisp Reference Manual, which will include material from the Interlisp-D users guide. This will be available soon in the fall of 1983; until then, the following documentation files are available now on [phylum]<Lisp>Fugue>:

ReleaseMessage.press/txt (this message)
PROC.<<<press/>>>tty
CHAT.<<<press/>>>tty
NEWETHER.<<<press/>>>tty
NEWARITH.<<<press/>>>tty
NS.<<<press/>>>tty
PROMPTFORWORD.<<<press/>>>tty
DURATION.<<<press/>>>tty
DEDIT.<<<press/>>>tty
TTYIN.<<<press/>>>tty<<<
1108OperatorGuide.press/tty>>>

As usual, please submit all problem reports, questions, to LispSupport.PA.

----

[1] Multiple 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. Users MUST read the current version of [phylum]<Lisp>Fugue>PROC.PRESS, for important instructions on living with the process world. For those users familiar with the changes to PROC in the Fugue.0 release, a summary of the major changes since then is herewith presented.

> Global input/output state 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. In Fugue.2, these 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 default; NIL defaults 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 are still fielded by the TtyProcess. However, BREAK (^B), 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] Xerox Network Systems protocols for filing supported, in addition to printing

Fugue supports a preliminary version of NS filing to Xerox 8030 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 must contain the ClearingHouse domain and organization, but if not supplied directly, then they are obtained from the defaults, which themselves are found by a search for the nearest ClearingHouse.

NS file servers are modeled after the Star world, and have "File Drawers" rather than directories; "File Folders" are like sub-directories. The functions DIRECTORY, FILEBROWSER, INFILE, COPYFILE, LOAD, and MAKEFILE are working now with NS file servers.

Support for Interpress-82 output (for Xerox 8040 printers) includes transmittal of screen images with scaling and rotation, listings with multiple fonts, lines and graphics; in general, hardcopy files may be in either INTERPRESS or PRESS format.

```
   (PRINTERMODE  'INTERPRESS)
```

changes the default for LISTFILES and hardcopy of windows and screen images to use Interpress format, and to send to Interpress printers. Output goes to NS.DEFAULT.PRINTER, if non-NIL, or else the first local printer it finds in your closest clearinghouse (see documentation on Clearinghouse protocols).

```
   (PRINTERMODE  'PRESS)
```

changes the default to PRESS. [Other printer modes are possible; for example, the PRINTER LispUsers package includes support for a low-cost dot-matrix printer connected through the parallel port.] Currently, the interface is not smart enough to infer the mode from a previously formatted file or the name of the host -- you must explicitly set the mode if you are sending a previously formatted (.PRESS or .INTERPRESS) file.

Fugue includes full support for standard Xerox Network Systems (NS) protocols, including SPP, Courier, Clearinghouse, Printing, and Filing. See preliminary documentation [Phylum]<Lisp>Fugue>NS.press/tty for more details on SPP, Courier, and Clearinghouse protocols.

----

[3] Dandelion's floppy disk drive -- Interlisp can read from it

Interlisp-D can read from the floppy drive, which is device {FLOPPY}. The functions DIRECTORY, FILEBROWSER, INFILE, COPYFILE, LOAD, READ, SETFILEPTR, etc. work now, with support for writing and for CP/M formatting to come later.

The actual floppy format is compatible with Pilot's usage.

Quite likely, we will be sending some update files via these floppy disks.

----

[4] TeleRaid: a remote debugging aid in lieu of Raid

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. 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 command key (bottom blank, or STOP on the 1108) 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).

----

[5] 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).

----

[6] EDITEMBEDTOKEN is now "&", affects both DEDIT and EDITF.
[Formerly, it was "*" for Teletype-style editor, and <esc> for DEDIT.]

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 * for the Teletype-style editor, and <esc> for DEDIT). Users are, of course, free to change the value in their init files.

----

[7] 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. PROMPT#FLG used to be initialized to NIL.

----

[8] Miscellaneous new variables:

NETWORKOSTYPES, MAX.SMALLP, MIN.SMALLP, MAX.FIXP, MIN.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)).

New global variables MIN.SMALLP, MAX.SMALLP, MIN.FIXP and MAX.FIXP are respectively the min and max on the range of the functions SMALLP and FIXP.

----

[9] RESTART.ETHER -- function to re-initialize ether drivers.

(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.

----

[10] New capabilities for RS232 users:
Functions for control of modem lines, RS232READSTRING for "fast" input
Asynchronous input buffering at slow speeds
"Roll" mode for RS232CHAT window, a {RS232} device interface

The RS232 driver file now supports control of a modem with the functions

(RS232MODEMCONTROL signalslst) [NoSpread Function]
Sets the modem control lines have on the signals in the list "signalslst"; returns the former setting of the lines. If "signalslst" is not supplied [which is not the same as supplying NIL], then the control lines are merely returned. Current signal names usable are DTR and RTS.

(RS232MODIFYMODEMCONTROL signalsonlst signalsofflst) [Function]
Changes only those modem control lines specified in the union of the two arguments; those in "signalsonlst" are set to be on, and those in "signalsofflst" are set off. Returns the former state just as (RS232MODEMCONTROL) does.

(RS232MODEMSTATUSP booleanform) [Function]

Returns non-null iff the reading of the modem status lines is consistent with the form "booleanform" [modem status signals currently supported are CTS, DSR, RI, and RLSD]. "booleanform" may be any AND/OR/NOT combination over the signal names. For example, (AND CTS (NOT RLSD)).

(RS232MODEMHANGUP) [Function]
Takes whatever steps appropriate to cause the modem to "hang up" [mostly, this means turning the DTR signal down for about 3 seconds, or until the DSR signal has gone down].

(RS232READSTRING #chars.limit? stopcode? noblocksflg wait? timerUnits oldstrbuffer)
RS232READSTRING will take input bytes from the RS232 port until one of three conditions obtains. (1) the total number of characters taken in by this call is equal to "#chars.limit?" [NIL means no limit]; (2) a character is read with character code equal to the argument "stopcode?" [NIL means no limiting charcter]; or (3) an interval of time greater than that specified by "wait?" has passed with no bytes available at the port. If "wait?" is non-null, it must be an integer, and "timerUnits" specifies the units (see the DURATION documentation). If "noblocksflg" is non-null, then RS232READSTRING will consume all the CPU cycles without offering to yield to other processes [including the MOUSE process]; this mode is important to very-time-critical applications. If "oldstrbuffer" is supplied, it must be a string and the result characters are smashed into it [so that no consing is done].

(RS232BACKGROUND state) [Funciton]
The "state" argument must be among {OFF, INPUT, OUTPUT, BOTH, ON}; except for input of OFF, this turns on a background low-level process which will service the UART at least once every 16 milliseconds and/or flush out the output buffer. A "state" of OFF shuts this background activity off. A "state" of INPUT causes only the input buffer to be serviced; OUTPUT for only the output buffer, and either ON or BOTH for both buffers. A period of 16 milliseconds for the input service time should give the appearance of asynchronous buffering, without dropping any characters, when used at speeds of less than about 600 baud.

RS232CHAT will pay attention to the DSPSCROLL setting of the chat window, and will do "roll" mode rather than "wrap" mode provided that it can do so without dropping characters ["roll" describes the "scroll up" action when typeout reaches the bottom of the window]. If the XON/XOFF protocol is being used, or if the background process mentioned above is in operation, then likely there will be no problem is using "roll" mode.

The {RS232} device is created by RS232INIT; one can obtain a stream interface to the RS232 port by calling (GETSTREAM '{RS232} <direction>). However, in most cases, this stream approach will not work unless the asynchronous buffering mentioned above is successful -- the time taken by general I/O operations is unpredictable and often quite large.

Ordinarily, a host will send a CR/LF for "newline", but some send only one; the menu selection lets you pick one or the other if this is the case (especially useful with UNIX systems). Similarly, you can specify that the RETURN key send either just CR or both CR/LF.

Several more implementations of the MODEM protocol for FTPing are available: one for the IBM/PC is available on floppy disk through XSIS (and is also on [MAXC]<XEOS>IBMFTP.ASC). Several files are available also on [MAXC]<XEOS> for VAX/VMS users: XMODEM.FOR and QIO.DCK are an implementation in FORTRAN; TOXMOD.FOR and FMXMOD.FOR are helpful for dealing with the structure of files in VMS's record management system.

Revised documentation is on [Phylum]<LispUsers>RS232.press/tty.

The remainder of this message recapitulates changes made in Fugue.0.

----

[11] Fugue runs compatibly on all three machines of the 1100 family:
  The 1100 (Dolphin), the 1108 (Dandelion) and the 1132 (Dorado)<<<
  An operators' guide for the Dandelion is available>>>

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. The 1108 is booted from the disk by a slightly generalized Initial microcode. 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.

For help in getting started, see the documentation on the file [Phylum]<Lisp>Fugue>.press/tty.

----

[12] 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 deleted 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 displaystream or window, 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.

----

[13] DEDIT in standard loadup

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 of chapter 9 of the Reference Manual. STANDARD will also be accepted for a while.

NOTE: The value of EDITEMBEDTOKEN, used by both Dedit's Replace command and the teletype editor's MBD command is now & by default. See item [6] of this note.

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 particularly choice pieces of text 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. 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. These various changes allow the user to write arbitrary programmed transformations to use under DEDIT since the full programmable power of the TTY editor is available.

Facilities now exist for users adding commands to and reorganizing the edit command window. Details on request.
----

[14] TTYIN installed as default linebuffer editor

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". Documentation for TTYIN is available on [Phylum]<Lisp>Fugue>TTYIN.tty/press. It may someday become obsolete when a more general text editor is completed and installed.

----

[15] 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. The DIR command, DIRECTORY, etc. all work for other partitions.

However, SYSOUT does not yet work for partitions other than the default.

----

[16] PupFtp protocol now supported; DIRECTORY works on remote file servers
Leaf protocol re-implemented with more flexibility

Fugue includes an implementation of the PupFtp protocol, which supports transferring files sequentially only. In those cases where sequential access (as opposed to random 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. The system tries to use PupFtp where possible for SYSOUT and for the destination file of a COPYFILE. One can indicate that a file is going to be accessed only sequentially by including the keyword SEQUENTIAL in the list of MACHINE.DEPENDENT.PARAMETERS passed to OPENFILE; the PupFtp will be used, if possible. This release does not support having PupFtp as the sole protocol supported by a file server; the necessary integration will occur in the next release. If for some reason your file server supports PupFtp but you do not wish COPYFILE or SYSOUT to use it, you can set the internal variable \FTPAVAILABLE to NIL.

The DIR command, and the DIRECTORY and FILDIR functions now work for files on remote servers (by using PupFtp) as well as on the local disk.

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.

----

[17] 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. For details, see <Lisp>Fugue>NewEther.press/tty.

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).

----

[18] System-level LispUsers packages now in standard loadup
DURATION (functions SETUPTIMER and TIMEREXPIRED? and new I.S.OPRS)
PROMPTFORWORD (formerly called READAWORD)

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. See the documentation on [Phylum]<Lisp>Fugue>DURATION.tty/press.

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. See the documentation file [Phylum]<Lisp>Fugue>PROMPTFORWORD.tty/press.

----

[19] 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). The following code equivalences are primarily for definitional purposes, and should not be considered an implementation (especially since the ireal implementation tends to be faster and less "consy" than apparent here).

```
   (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 compare 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 any 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 fieldsize is WORD and n is 1.

----

[20] 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.

----

[21] New and revised LispUsers packages; some incompatible between Chorus and Fugue

Please note that new or newly revised LispUsers packages have only been verified to run under Fugue. Some of these may not work correctly when loaded into older releases. Conversely, older versions of LispUsers packages may not work correctly (or even crash the system) when loaded into the Fugue release. As always, there are many revised <LISPUSERS> packages, not synchronized with the release.

The following packages are known not to be compatible between Chorus and Fugue, with new version available on [Phylum]<LispUsers>:

PRESSBITMAP
EVALSERVER
all RS232 files
RPC (The Cedar remote procedure protocol)
LAFITE, GRAPEVINE, MAILCLIENT (for electronic mail files).

Old versions of these packages will not work correctly in Fugue, but will be preserved on [Phylum]<LispUsers>Chorus> for a period of time.

The following packages are included in the standard loadup and thus need not be loaded; in most cases the old versions are incompatible with Fugue:

ADDARITH
BSP, CHAT
CHARCODE
DEDIT, DSPRINTDEF, NEWPRINTDEF
DURATION
PROC
READAWORD
GLOBALRESOURCE

The following have appeared recently:

EDITBITMAP - More bitmap manipulations a la Notepad
EDITMACROS - some (TELETYPE) editor macros
FILEBROWSER - A menu-driven interface to a file structure;
Main entry is function FILEBROWSER
LISPUSERS.TTY - an index of all LispUsers packages, with brief description of each.
LOGOCLOCK - A clock process in the Interlisp-D window
MAKEGRAPH - produces forests by walking over data structures
according to a specification for that walk
NQUEENS - solves the n-Queens problem graphically
PRINTER - LISTFILES and HARDCOPYW (of windows, bitmaps, etc.)
on C.Itoh and other low-cost graphics printers.
RPC - Lisp version of Cedar MESA's remote procedure call mechanism
SAMPLER - graphics demos
STOCKICONS - use in conjunction with ICONW
STYLESHEET - coordinated selections from a set of menus

SIGNAL - Lisp version of Cedar MESA's signal mechanism
TRAJECTORY-FOLLOWER - causes a 'snake' to crawl along a curve.

The following have been recently revised:

EVALSERVER	- 18-JUN-83 (Fugue-only version)
BIG	- 29-JUL-83
BITMAPFNS	- minor fixes
TEDIT	- greatly revised version.
TEXTOFD	
TEDITHCPY	
CMLARRAY	- 2-JUL-83
COMPILEBANG	- 10-MAR-83
COREHAX	- 3-JUN-83
CROCK	- 25-JUL-83
HANOI	- 2-JUN-83
HEADLINE	- 11-MAY-83
HISTMENU	- 31-MAR-83
ICONW	- 9-AUG-83
LLCOLOR	- 6-JUL-83
NOTEPAD	- 11-AUG-83
POLYGONS	- 5-MAY-83
PROMPTREMINDERS	- 25-JULY-83
TMENU	- 5-JUN-83
UTILPROC	- 7-MAR-83

---

[22] Known problems with this release

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 technically incorrect 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.

If you LOGOUT while some other process is actively writing a file, it is not guaranteed that anything reasonable happens to the file.

There is no utilization of the 1108's rigid disk for the {DSK} device -- it is only "ephemeral" and files placed there will not survive a re-loading or re-booting. Later this fall, full support, with directories too, will be announced.

The swapping file (Lisp.VirtualMem on 1100s and 1132s) must still be a "small" number of disk sgements. 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.
