

Extracting the data from this tape:

Most of the data on this tape has been archived to reduce the number of files so that it will write this tape. You should extract the data from the libraries. This will require about 10000 blocks of storage if you don't remove each "cont.a" after you extract it.

What is on this tape:

This tape includes sources and binaries for a quantity of UC Berkeley software. The major items on this tape are the instructional Pascal system and the text editor "ex". Other software here includes a modified shell, a new shell, new commands, and a "star trek" game. Machine readable documentation is included for all programs. The "Pascal User's Manual" and the "Ex reference manual" need a phototypesetter if readable copies are to be produced.

Pascal:

The Pascal system has been used for a year for instruction here. It has undergone a number of improvements in the last six months, but is quite stable. We use it for undergraduate and graduate instruction.

The Pascal system requires separate I/D space; that is, an 11/45 or 11/70 host. To run the Pascal stuff right away you will also need floating point hardware -- it is possible to run Pascal without floating point hardware but it requires adding a system call to replace the "mfpi" instruction that doesn't work on the 11/45's and 11/70's in user mode.

Ex:

The Ex editor has been in use for about the same length of time as Pascal, and is used by a majority of our users. It has undergone a number of improvements in the last few months. We intend to use ex for the text editing classes at the Computer Center here (for a general campus audience) starting in January. The Pascal documentation uses "ex" in its examples.

Installing the software:

Compiled binaries have been included for most of the software here. (A few of the routines in the directory s6 include system dependent headers and so binaries would be of no use and are not included.)

The major programs "pi", "pxp", "px", and "ex-1.1" have the binaries in the directories with the same names. "Pi", "pxp" and "px" should run as they stand... if you have a non-standard teletype driver "ex" may require some system dependent changes. The binaries in "ex" will run directly on standard or Berkeley-type version 6 UNIX systems.

Each major directory includes a file "READ_ME" describing the software in the directory. There is often a shell script "setup" in the directory to perform one time only operations. The script "install" in these directories will place the software in its standard home.

For recompilation of these programs you can use the scripts "make*", and "comp" and "load" in the directories. Most directories also have "print" scripts, i.e. "printpi", to make a program listing with utility files and programs in a reasonable order.

The suggested way to bring up the software on this tape is to run the install scripts in "pi", "pxp", and "px", and to then install (some or all) of the software from "bin". The editor "ex" can also be installed... this requires probably as much work as all the others combined as it uses some data bases which don't exist on standard UNIX relating to terminal types and capabilities.

Note that some of the scripts to make new versions of the software on this tape use the programs in "bin". You can run these scripts easily, without adding all the stuff in "bin" to your "/usr/bin", by using the shell in

"ashell". This shell has a number of nice features and was used in making all the software here... the files "errs" in each of the major directories are outputs from the "make*" scripts so you can see how this was done. Documentation for "ashell" is in s6/sh.6.

The trek game in "trek" uses the rewritten portable library in "portlib". It (and the program tset) were written by Eric Allman whose address is in the trek setup instructions in "trek".

Directory contents:

```
lw(10) l.
pi    Pascal translator source
px    Pascal interpreter
pxp   Pascal execution profiler
eyacc Modified yacc for Pascal
assubs      Assembly stuff for Pascal
tests  Test programs for Pascal
pcs    Wirth's Pascal-S
pxref  Pascal cross-reference program
opcodes  Definition files for Pascal
fpterp Sep ID floating point interpreter using FETCHI sys call
s?     Command software source
man?   Documents for s? stuff
ashell A new shell with some nice features
ex-1.1  Ex source
exrecover Ex recovery routines (after system crashes)
trek   Source for a "star trek" game
portlib  Portable library used by trek
exrefm  Troff source for "Ex 1.1 Reference Manual"
puman   Troff source for "UNIX Pascal User's Manual"
help   Sections from our help command

lib    Routines for /lib and /usr/lib
bin    Routines for /usr/bin
etc    Stuff for /etc
```

If you don't have floating point:

If you don't have floating point hardware, and wish to run Pascal, you will need to add a system call to fetch an instruction word when running separate I/D so that the floating point interpreter can work. The system call to be added is "fetchi"... if you can make it system call 61. (decimal) then the binaries on this tape will work immediately. The code for "fetchi" will reside in sys4.c and look like

```
fetchi()
{
    u.u_ar0[R0] = fuiword(u.u_ar0[R0]);
}
```

It is used as in:

```
mov    $iaddr,r0
sys    fetchi
```

to get the contents of location "iaddr", a word in I-space. Look at the floating point interpreter in the directory fpterp for a sample. Don't forget that to make the system call work you must add an entry to the syent array in sysent.c.

Miscellaneous notes:

A version 7 C compiler and many of the binaries in "bin" are required to make a new version of "pascal" or "ex". For Pascal the file "nofloat" in this directory should exist... it causes the scripts in the source directories to work slightly differently.

Feedback:

We would like to hear from all users of the Pascal system and of ex. Reports of problems in bringing this software up, or of bugs in the programs or documentation would be appreciated. We would also appreciate hearing of any local improvements you make.

11/34 or 11/40 Pascal:

It is indeed unfortunate that the Pascal system here won't run on an 11/34 or 11/40... the only reason this is true is that the translator "pi" is too large. Pi used to be two pass but was made one pass about a year ago. It is certainly possible to break pi into two passes or two processes communicating through a pipe. If you have an 11/34 or 11/40 and are interested in trying this I will be glad to give you more details.

Bill Joy
CS Division
Department of EE and CS
UC Berkeley
Berkeley, California 94704

(415) 524-4510 [HOME]
(415) 642-4948 [SCHOOL]