my file

SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED INTERACTIVE COMPUTING NOTE 726

PERQ UNIX IMPLEMENTATION NOTE # 30
Changes to UNIX Include Files

Issued by
C Prosser

25 November 1982

---

DISTRIBUTION:
R W Witty
K Robinson
C Prosser
A S Williams
L O Ford
T Watson
E V C Fielding
J C Malone
P J Smith
J M Loveluck
RL Support/PERQ/Unix Implementation Notes file

## 1. Introduction

This document describes the reasons for changes to UNIX include files (ie: those files residing in the subtree /usr/include) for the ACCENT based UNIX system running on PERQ (up to release 0.3 base version). The UNIX include files, together with the UNIX system call specification and definitions of UNIX file formats define the application programmer's interface to the UNIX system. Unless otherwise stated, the generic term UNIX is used throughout to refer to Version 7 UNIX. Definitions are given preferentially in terms of the C programming language in keeping with the conventions of the UNIX Programmer's Manual. Information about 3RCC Pascal files is given where appropriate. Files which reside in the /usr/include subtree but whose usage is mostly restricted to implementation of system support facilities, such as file servers and the like, are described in separate documents.

References of the form ar(5) are to items documented in the UNIX Programmer's Manual Volume 1. References of the form [3] are to documents published elsewhere.

## 2. Status of Include Files

The following tables indicate which files have been changed, introduced or deleted from the original Bell Labs distribution of UNIX (December 1979). Subsequent sections describe reasons for amendments in detail. Where a 3RCC Pascal equivalent exists, its base name is the same as that of the C language version but with suffix ".dfs" rather than ".h".

Files in /usr/include:

| Name | Status | 3RCC Pascal Equivalent |
|---|---|---|
| a.out.h | changed | yes |
| ar.h | changed | yes |
| assert.h | same | no |
| code.h | new | yes |
| core.h | same | no |
| ctype.h | changed | no |
| dk.h | same | no |
| dumprestor.h | same | no |
| errno.h | changed | yes |
| execargs.h | changed | yes |
| grp.h | same | no |
| ident.h | changed | no |
| math.h | changed | no |
| mp.h | same | no |
| olddump.h | same | no |
| pack.h | same | no |
| pwd.h | same | no |
| ranlib.h | new | yes |
| saio.h | same | no |
| setjmp.h | changed | yes |
| sgtty.h | changed | yes |
| signal.h | changed | yes |
| stdio.h | changed | yes |
| symbol.h | changed | no |
| sys.s | deleted | no |
| time.h | same | no |
| tp_defs.h | same | no |
| utmp.h | changed | no |
| varargs.h | changed | yes |
| whoami.h | same | no |

Files in /usr/include/sys:

| Name | Status | 3RCC Pascal Equivalent |
|---|---|---|
| dir.h | changed | yes |
| oboe.h | new | no |
| param.h | changed | yes |
| pstring.h | new | no |
| stat.h | changed | yes |
| timeb.h | same | yes |
| times.h | changed | yes |
| types.h | changed | yes |

The following files are deleted from /usr/include/sys as they have no meaning or use in an ACCENT based UNIX system:

```
acct.h      buf.h      callo.h   conf.h    fblk.h    file.h
filesys.h   ino.h      inode.h   map.h     mount.h   mpx.h
mx.h        pk.h       pk.p      prim.h    proc.h    reg.h
seg.h       systm.h    text.h    tty.h     user.h
```

The following 3RCC Pascal include files have been introduced in /usr/include/sys:

```
Ctype.dfs   Ptype.dfs   acb.dfs   eeb.dfs   except.dfs   mch.dfs
ps.dfs      rd.dfs      vrd.dfs
```

As these latter files are intended for use by system programs and specialised application programs written in 3RCC Pascal there are no corresponding C files.

## 3. Reasons for Changes

Changes have only been introduced where absolutely essential due to differences in architecture between PDP-11 and PERQ (eg: object file format), or fundamental design differences in the system from "standard" UNIX (eg: physical filesystem format). The intention is to retain compatability with standard UNIX wherever possible, thus minimising changes to programs which include changed header files. However, potentially misleading references to features not implemented (by design) have been deleted (eg: see entry for stat.h below). Where feasible existing names of identifiers have been re-used for directly corresponding items and ordering of fields in structures has been retained. For a complete understanding of the alterations described it will be necessary to consult the source code (available on-line on PDP-11/70 under the subtree /usr/spice/curator/accunix/usr/include). Changes to the aesthetic layout of text which do not affect the semantic meaning (eg: extra blanks) are not recorded here.

In several places built-in constants have been parameterised. It is recommended that programs should use the appropriate parameter rather than follow the historical practice of binding arcane knowledge of system constants into source text.

| File | Remarks |
|---|---|
| a.out.h | This file has been extensively altered to cater for the redesigned object file format (see a.out(5)). Further details of changes are given in [11]. |

File Remarks

ar.h

The following changes have been made

a) The formerly built in constant size of 14 characters for an archive module name has been parameterised as

```
#define ARNAMSIZ (25+1)
```

On most systems the value chosen corresponds to the value of DIRSIZ (see entry for dir.h below).

b) The type of the ar_date field has been changed from long to time_t to correspond with the usual definition of the time type in types.h.

c) The ar_uid and ar_gid fields have been redeclared as short to correspond with normal usage in the rest of UNIX (ie: 16 bit quantities; but perhaps they should really be declared int)

d) The line

```
typedef unsigned short armag_t;
```

has been introduced to make explicit the type of the magic header pattern in archives. The (unchanged) definition of ARMAG is, by default, of type int. By convention the values of magic header patterns occupy 16 bits, the size of a short item on PERQ [13]. Equality comparison of the integer ARMAG with a signed short quantity containing the same bit pattern will fail since, by definition of C, shorts are converted to integer with sign extension before the comparison is performed.

(see also ar(5))

code.h

This file has been introduced to aid programs processing the SEG files produced by 3RCC Pascal. For an explanation of SEG file format see [4].

ctype.h

A minor change has been made to a comment which referred to the non-existent file

/usr/src/libc/gen/ctype_.h

instead of

/usr/src/libc/gen/ctype_.c

File Remarks

errno.h

The following changes have been made

a) The error numbers 1 through 34 inclusive have unchanged definitions and meanings as described in intro(2). Brief explanatory comments have been placed alongside the corresponding constant definitions as an aid to system maintainers. The following additional error numbers have been defined (for full definition of meanings see intro(2)):

```
#define ENOPORT  35  /* no ports left */
#define EPTOFLO  36  /* port table overflow */
#define EPPOFLO  37  /* pipe table overflow */
#define EDEVOFLO 38  /* device table overflow */
#define EMTOFLO  39  /* mount table overflow */
#define EBADPT   40  /* bad port number */
#define EEOF     41  /* end of file */
#define EMATCH   42  /* matchmaker (IPC) interface error */
#define EACCENT  43  /* accent general error */
#define ENOKERN  44  /* not a kernel port */
```

b) A comment indicating that error numbers larger than 32 referred to maths software has been deleted.

execargs.h

This horrid trick played by sh(1) on PDP-11 to set up command line information for ps(1) is utterly banned on PERQ!!! See instead the descriptions of exec(2) and ps(2) implementations [14].

ident.h

Rather trivial change to a string constant identifying the current system, and surprisingly hard to remember to keep up to date. Not actually used by any standard utilities to the best of my knowledge.

math.h

See document describing changes made to floating point library routines [9].

ranlib.h

New file introduced to formalise the format of the special archive entry generated by ranlib(1).

(see also ranlib(5))

File Remarks

setjmp.h

The size of the array containing stack frame information which must be preserved to implement the non-local goto has been increased to accord with the requirements of the PERQ Q-code architecture [2]. Furthermore, as the primitive routines to save and restore the environment are actually implemented on PERQ as special system entries [14], rather than as directly callable assembly code routines, macro definitons for setjmp(3) and longjmp(3) have been introduced, which MUST expand to in-line calls on the primitive operations [14]. The user is required to include this file in any C module making use of the non-local goto facility.

sgtty.h

A fresh numbering scheme for TIO????? and FIO????? requests has been introduced for the benefit of 3RCC Pascal implementation. DIO????? and MX????? requests have been deleted as those facilities are not implemented. The retained TIO????? and FIO????? requests correspond to those in most common usage at sites running the EUUG V7 UNIX distribution. (See tty(4) for details of which requests are currently implemented).

signal.h

The following changes have been made

a) Two additional signal numbers have been defined, viz:

```
#define SIGMEMPAR  16  /* memory parity error */
#define SIGRANGE   17  /* expression out of range etc */
```

Accordingly the value of NSIG has been increased to 18. (Previously signal number 16 existed although unassigned).

b) The definitions of SIG_IGN and SIG_DFL as coercions on integer values have been replaced by definitions which reference genuine functions which have been implemented as special system entries [14]. It is, therefore, not meaningful on PERQ to treat return values from signal(2) as integer values, test for odd or even, or to pass integer parameters as function addresses. Non-portable programs which indulge in obscure implementation dependent practices MUST be changed. In a similar vein, the error return from signal(2) has been defined as SIG_ERR, which references a genuine function implemented as a special system entry. It is not meaningful to compare the return value of signal(2) with the integral value -1 when testing for the error "signal out of range".

File Remarks

stdio.h

The following changes have been made

a) The value of BUFSIZ has been changed to (8*512). Note that this value no longer happens to correspond to the value of BSIZE (see param.h below) so that programs such as sum(1) which assumed that they were the same MUST be corrected.

b) the fields _flag and _file of the _iob structure are logically integral values and their types have been changed to short (from char).

c) Line buffering à la Berkeley 4.1BSD has been introduced in the stdio package [7]. The concept has been extended to differentiate between interactive I/O attached to a terminal and I/O which happens to be line buffered. Thus two extra values for the _flag field have been defined:

```
#define _IOLBF  0400   /* line buffered I/O */
#define _IOTTY 01000   /* interactive I/O */
```

It is not expected that application programs will noramlly need to use these definitions. Note that _IOLBF and _IOTTY require the _flag field to be wider than 8 bits; beware extensions to the stdio package which have the _flag field declared inappropriately or have only partially correct handling of interactive I/O.

c) The macro definition of fileno(3) has been amended to cope with the possibility that its parameter may be an expression, viz:

```
#define fileno(p)  ((p)->_file)
```

(note parenthesised 'p' in expansion)

d) The missing declaration

```
FILE *popen();
```

has been inserted.

e) The macros for getc(3) and putc(3) have been enhanced to check for read and write access respectively on the streams given as parameters. This corresponds to what happens when the genuine function versions are used.

File Remarks

symbol.h

This alternate definition of object file symbol table layout to that in a.out.h has been changed to reflect the redesigned object file format. Note also the introduction of the defined constant

```
#define SY_NAMSIZ  8
```

to parametrise the number of characters in a symbol name. Care must be taken to ensure that the definitions in this file correspond to those in a.out.h.

(see also a.out(5) and [11])

utmp.h

The constant definitions

```
#define UT_NAMSIZ   8
#define UT_LINESIZ  8
```

have been introduced to parameterise the corresponding array sizes.

varargs.h

The macros and definitions have been extensively reworked in an attempt to make them even more machine independent than previously. It is a matter of personal opinion as to whether the desired objective has been achieved or not. For example usage see the source code for doprnt.c in the stdio library [7]. The definitions here are incompatible with "standard" V7 UNIX and Berkeley 4.1BSD UNIX, but, to the best of my knowledge, no "standard" V7 UNIX programs use them.

On PERQ, due to Q-code architecture limitations, there is a (generous) upper limit on the parameter space which may be used without losing data on a call to a function declared with a va_list parameter. Currently VA_MAX (=100) int's worth of space is available per va_list. The link editor ld(1) will warn about parameter data loss on function calls [8].

File Remarks

dir.h

The allowable length of a filename has been increased to match that currently provided by the Interim Spice Filesystem [3], viz:

```
#define DIRSIZ  (25+1)  /* name + null terminator */
```

A new library routine getdirent(3) has been provided for use of programs which wish to read directory files [5]. This routine provides a mapping between the physical disk format (for definition of this see description of Module DiskIo in [1]) and the format assumed by UNIX.

oboe.h

New file providing C external definitions of primitive system entry functions. Also provided are macro definitions to assist in implementing calling of 3RCC Pascal procedures and functions from C [10]. These macros convert C representation of character pointers to 3RCC Pascal representation and are commonly used in conjunction with the function P_MAKESTRING which is declared in the module pstring.c in the C run-time support library [6].

(see also [14,12] and pstring.h below)

param.h

All that remains of this file are definitions of machine dependent tunable parameters that apply to the PERQ, viz:

```
#define HZ        60    /* Ticks/second of the clock */
#define TIMEZONE  (0)   /* Minutes westward from Greenwich */
#define DSTFLAG   1     /* Daylight Saving Time applies locally */
#define BSIZE     512   /* disk block size */
#define BSHIFT    9     /* log2 (BSIZE) */
```

Note that macros which formerly appeared in param.h, but were also defined in types.h, are now defined only in types.h.

(see also types.h below)

pstring.h

New file introduced to define structure for converting from C style strings to 3RCC Pascal strings which is used in the module pstring.c of the C run-time support library [6]. For details of inter-language procedure calling conventions see [10].

(see also oboe.h above)

File        Remarks

stat.h

The definitions of multiplexed special file mode bits have been deleted as the mpx(2) facility is not provided in this implementation.

times.h

The type definition of structure members holding time information has been changed from long to time_t. This implies that any module using this include file must now include types.h.

(see also types(5))

types.h

The following changes have been made

a) As the PERQ Q-code architecture [2] is not register oriented the declaration of label_t has been deleted.

b) The device code type dev_t occupies 16 bits so the definition has been changed to short.

c) The constant definition

```
#define void int    /* NON-MEANINGFUL return value */
```

has been introduced. This is a rather pathetic attempt to make visible the fact that some C functions return, by design, random garbage, which should not be inspected. A better solution is to make the type void a primitive type of the C language.

Ctype.dfs

This file contains definitions in terms of 3RCC Pascal of the C language representations of various elementary data items.

(see also [10])

Ptype.dfs

This file contains definitions of various 3RCC Pascal representations of elementary data items intended to assist in the writing of portable code.

(see also Ctype.dfs and mch.dfs)

acb.dfs

This file is a copy of the corresponding system micro-code definition file for a procedure or function activation control block [2].

eeb.dfs

This file is a copy of the corresponding system micro-code definition file for an exception enable block [2].

File | Remarks

except.dfs

This file is a copy of the corresponding system micro-code definition file for symbolic names of microcode plus accent detected exception conditions [2].

mch.dfs

This file provides 3RCC Pascal definitons of PERQ machine representations of various quantities (eg: byte, word, array pointer etc).

ps.dfs

This file defines the structure filled in by the ps(2) system call [14].

rd.dfs

This file is a copy of the corresponding system micro-code definition file for a procedure or function routine dictionary used in POS SEG files and C object files.

(see [2,4,11] and a.out(5))

vrd.dfs

This file is a copy of the corresponding system micro-code definition file for a procedure or function variable routine descriptor [2].

REFERENCES

1. "Perq Operating System Interface", Perq Software Reference Manual Version D5, Three Rivers Computer Corporation (June 1981).

2. M.A. BAREL, J.P.STRAIT, S.P. HARBISON, AND G.G. ROBERTSON, Perq Q-code Reference Manual (Second Edition), Three Rivers Computer Corporation, and Carnegie-Mellon University, July 1981.

3. L.O. FORD, "Perq Unix Implementation Note # 5 - Comparison of UNIX and SPICE interim filesystems", DCS Note # 552, Rutherford Appleton Laboratory (January 1982).

4. E. FRANKENBERRY, PERQ Segment Files, Three Rivers Computer Corporation, July 1981 .

5. C. PROSSER, "Perq Unix Implementation Note # 34 - Changes to Gen Library", DCS Note # 730, Rutherford Appleton Laboratory (November 1982). [In Preparation].

6. C. PROSSER, "Perq Unix Implementation Note # 37 - Changes to Crt Library", DCS Note # 733, Rutherford Appleton Laboratory (November 1982). [In Preparation].

7. C. PROSSER, "Perq Unix Implementation Note # 33 - Changes to Stdio Library", DCS Note # 729, Rutherford Appleton Laboratory (November 1982). [In Preparation].

8. C. PROSSER, "Perq Unix Implementation Note # 36 - UNIX Linker Implementation", DCS Note # 732, Rutherford Appleton Laboratory (November 1982). [In Preparation].

9. P.J. SMITH, "Perq Unix Implementation Note # 32 - Changes to Math Library", DCS Note # 728, Rutherford Appleton Laboratory (November 1982). [In Preparation].

10. T. WATSON, "Perq Unix Implementation Note # 35 - Inter-Language Procedure Calling Conventions", DCS Note # 731, Rutherford Appleton Laboratory (November 1982). [In Preparation].

11. A.S. WILLIAMS, "Perq Unix Implementation Note # 24 - Perq UNIX object file format", DCS Note # 607, Rutherford Appleton Laboratory (May 1982).

12. A.S. WILLIAMS, "Perq Unix Implementation Note # 25 - Type mapping between C, Fortran 77, and Pascal", DCS Note # 608, Rutherford Appleton Laboratory (May 1982).

13. A.S. WILLIAMS, "Perq Unix Implementation Note # 22 - Mapping of C data types for Q-code", DCS Note # 605, Rutherford Appleton Laboratory (May 1982).

14. A.S. WILLIAMS, "Perq Unix Implementation Note # 31 - UNIX System Call Specification", DCS Note # 727, Rutherford Appleton Laboratory (November 1982). [In Preparation].