SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED INTERACTIVE COMPUTING NOTE 789

PERQ UNIX IMPLEMENTATION NOTE # 57
Accent/Unix: Generr Program

Issued by
J.C.Malone

26 January 1983

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
A J Kinroy
J M Loveluck
C P Wadsworth
Martin Ritchie (ICL Dalkeith)
P Palmer (ICL Dalkeith)
RL Support/PERQ/Unix Implementation Notes file

The Generr program is written in C to run on any Unix implementation. It tests the results returned and error number set for the following system calls [3] :

Chdir	Chmod
Chroot	Close
Creat	Dup
Fstat	Ioctl
Link	Lseek
Mknod	Mount
Open	Pipe
Read	Stat
Umount	Unlink
Utime	Write
Exec	Kill
Signal	Wait

Required files

Certain files are used, of which the characteristics are known, e.g. an existing directory, a non-existent file. To ensure that these are prepared, the shell command file, setup, is used to create a temporary directory containing these files and to run the program in there.

The following names give the type of file for which the system call is being tested:

FILENOEXIST - non-existent, but legal, filename
PATHNOEXIST - one of subdirectories in pathname does not exist
FILEEXIST - existent file
PATHNODIR - one of subdirectory names is not a directory
NONEXEC - existent, but non-executable file
ANYNAME - legal filename, do not care if exists
DIREXIST - existent directory
NOTTY - existent special file, but not /dev/tty
BLOKDEV - mountable file system
MOUNTEDFILE - existent file on mounted file system
MOUNTNEWFIL - non-existent file on mounted file system

Results on AccUnix

The output shows the types of file for which the results or error numbers differed from those expected. Those system calls which were successful are not printed.

The label at the beginning of each line refers to the sections which follow to explain the output.

```
A1 chdir(PATHNODIR)                 expected: ENOTDIR  errno:  ENOENT
A1 chmod(PATHNODIR,00000)           expected: ENOTDIR  errno:  ENOENT
A1 creat(PATHNODIR,000666)          expected: ENOTDIR  errno:  ENOENT
A2 creat(NULLNAME,000666)           expected: ENOENT   errno:  EISDIR
A1 link(PATHNODIR,FILENOEXIST)      expected: ENOTDIR  errno:  ENOENT
A5 link(DIREXIST,FILENOEXIST)       expected: EPERM    errno:  EISDIR
A1 link(FILEEXIST,PATHNODIR)        expected: ENOTDIR  errno:  ENOENT
A2 link(FILEEXIST,NULLNAME)         expected: ENOENT   errno:  EEXIST
A1 mount(PATHNODIR,DIREXIST,RO)     expected: ENOTDIR  errno:  ENOENT
B1 mount(FILEEXIST,DIREXIST,RO)     expected: ENXIO    errno:  ENOENT
A1 mount(BLOKDEV,PATHNODIR,RO)      expected: ENOTDIR  errno:  ENOENT
A1 mount(BLOKDEV,FILEEXIST,RO)      expected: ENOTDIR  errno:  ENOENT
A1 open(PATHNODIR,RW)               expected: ENOTDIR  errno:  ENOENT
A1 stat(PATHNODIR,&statbuf)         expected: ENOTDIR  errno:  ENOENT
A1 umount(PATHNODIR)                expected: ENOTDIR  errno:  ENOENT
B1 umount(FILEEXIST)                expected: ENXIO    errno:  ENOENT
A1 unlink(PATHNODIR)                expected: ENOTDIR  errno:  ENOENT
A3 utime(FILENOEXIST,timep)         expected: -1       result: rubbish
A3 utime(FILENOEXIST,timep)         expected: ENOENT   errno:  rubbish
A3 utime(PATHNODIR,timep)           expected: -1       result: rubbish
A3 utime(PATHNODIR,timep)           expected: ENOTDIR  errno:  rubbish
A3 utime(PATHNOEXIST,timep)         expected: -1       result: rubbish
A3 utime(PATHNOEXIST,timep)         expected: ENOENT   errno:  rubbish
A1 execve(PATHNODIR,&argv,&envp)    expected: ENOTDIR  errno:  ENOENT
A4 execve(DIREXIST,argv,envp)       expected: EACCES   errno:  rubbish
A4 execve(NONEXEC,argv,envp)        expected: EACCES   errno:  rubbish

Super-User calls:
A1 chroot(PATHNODIR)                expected: ENOTDIR  errno:  ENOENT
A1 mknod(PATHNODIR,FIFOSP,dev)      expected: ENOTDIR  errno:  ENOENT
A2 mknod(NULLNAME,FIFOSP,dev)       expected: ENOENT   errno:  EEXIST
```

A. Explanation of program results

1 SwitchBoard [1] returns ENOENT when a file cannot be found, or cannot be created.

There is no distinction between:
ENOENT - file or subdirectory does not exist
ENOTDIR - subdirectory is not a directory

2 A file with a nullname is passed to SwitchBoard as the current directory, so errors generated will be those for an existing directory, and not specifically for a nullname.

3 UTime - ErrNo is not set, since SysUTime returns the type P_Void, which is ignored (hence rubbish result).

4 ExecVe - ErrNo is not set for EACCES, since the function which checks whether a file is executable returns an error value which is lost.

5 Even the Super-User cannot link to a directory, so the error EPERM would be misleading and EISDIR is used instead.

B. Further Tests on AccUnix

Features peculiar to AccUnix will need to be tested separately:

1 Mount and UMount [3,2] expect POS-style partition names.

So the error will not be as expected in generr with FILEEXIST.

Instead, try

```
mount "sys:user>guest" /tmp
umount "sys:user>guest"
```

to get ENXIO (6) No such device or address.

2 Linking across partitions:

mount sys:user /tmp

link <exists in wd> <doesn't exist in /tmp>

to get EXDEV (18) Cross-device link.

REFERENCES

1. J.C. MALONE, "Perq Unix Implementation Note # 41 - Accent/Unix: Switchboard", DIC Note # 744, Rutherford Appleton Laboratory (January 83).

2. J.C. MALONE, "Perq Unix Implementation Note # 51 - Accent/Unix: Special Files, Mountable File Systems", DIC Note # 756, Rutherford Appleton Laboratory (January 83).

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