



OpenVMS UNIX
Application Portability
Initiative

Brad McCusker
Transition Engineering & Consulting,
Business Critical Servers


© 2005 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



Topics

- Introduction to UNIX Portability
- OpenVMS V8.2
- Future Releases
- Summary

20-Jan-2003 Partners Presentation 2003 2




Various Terms – all mean the same...

- UNIX Portability is the original term used to describe this effort
- The effort quickly evolved to include:
 - Linux Interoperability
 - Linux Portability
 - Open Source Interoperability
 - Open Source Portability

All these are part of UNIX Portability


20-Jan-2003 Partners Presentation 2003 3



Unix Portability - Rationale

- Many ISVs develop applications for both OpenVMS and UNIX/Linux platforms
- Applications are (or can be) ported from UNIX/Linux platforms to OpenVMS
- Operators, programmers, users may be more familiar with *NIX-style interfaces, commands, utilities and tools

20-Jan-2003 Partners Presentation 2003 4




UNIX Portability - Goal

Provide a full set of UNIX interfaces and tools within OpenVMS

- In native, integrated fashion
- No layered emulator (e.g. old “POSIX for OpenVMS” product)
 - No performance issues
 - No interoperability issues


20-Jan-2003 Partners Presentation 2003 5



UNIX Portability - Benefits

- Easy portability of UNIX applications to OpenVMS
- Easy development of applications intended to run on both UNIX and OpenVMS
- No need to train UNIX-skilled personnel on OpenVMS
- OpenVMS will optionally be like a “UNIX flavor”
 - Cost of porting from UNIX to OpenVMS equal or comparable to porting from one “UNIX flavor” to another (e.g. from Solaris to Tru64)


20-Jan-2003 Partners Presentation 2003 6



But – I like VMS the way it is!!!

- Current VMS behavior is preserved
 - New UNIX Portability features typically need to be enabled
 - Defaults preserve existing behavior
- C Run Time Library: UNIX features are enabled via logical name switches
 - Old behavior is the default
 - Legacy behavior is preserved
 - Can also enable features via an API


20-Jan-2003 Partners Presentation 2003 7



Rollout...

- Started already with VMS V7.3-1, V7.3-2, V8.2 ...
 - GNV/BASH (Commands & Utilities)
 - C RTL
 - New UNIX APIs
 - Improved UNIX filename support
 - API for controlling feature switches
 - File system improvements
 - Mixed case file names, case sensitive compares
 - Time of last file access
 - Symbolic links (coming soon...)
 - Shared stream I/O (investigation)
 - Base VMS improvements
 - Extended DCL line length


20-Jan-2003 Partners Presentation 2003 8



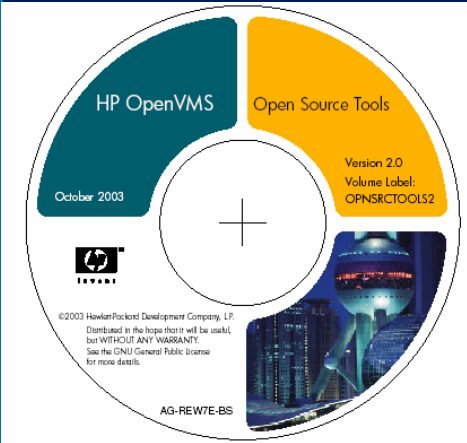
GNV

- GNV: GNU's Not VMS
 - GNU-based, UNIX® environment for OpenVMS
 - Open source, freeware product
 - <http://gnv.sourceforge.net/>
 - Implementation of the UNIX shell BASH (Bourne Again Shell)
 - Provides an environment for porting and running UNIX tools and software on OpenVMS
 - Updates include:
 - ODS-5 file system support
 - Additional utilities ported and included
 - Packaged as a HP-branded PCSI kit

20-Jan-2003 Partners Presentation 2003 9




Open Source Tools CD



- Ships with OpenVMS V7.3-2
- Includes GNV 1.6-2
- Other contents:
 - Stunnel 3.26
 - GnuPG 1.23
 - GTK+
 - libIDL
 - CDRECORD V1.10
 - Sources
 - SSL 1.1-A
 - CDSA 2.0
 - Kerberos 2.0
 - OpenVMS Migration Software


Updates at:
<http://h71000.www7.hp.com/opensource/opensource.html>

20-Jan-2003 Partners Presentation 2003 10




What we've done...

UNIX Portability Features



© 2005 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



V7.3-2 C RTL Contents

- New APIs
 - User database functions
 - getpwnam_r, getpwuid_r
 - Signal functions
 - sigwait, sighold, sigrelse, sigignore
 - Clock functions
 - nanosleep, clock_getres, clock_gettime, clock_settime
 - Math and conversion functions
 - rand_r, remainder, rint, a64l, l64a
 - Pattern matching functions
 - glob(), globfree()

20-Jan-2003 Partners Presentation 2003 12



V7.3-2 C RTL Contents (cont)

- New APIs (cont)
 - Security/Impersonation Functions:
 - endgrent getgrnam getsid setpgrp
 - getgrent getgrnam_r seteuid setregid
 - getgrgid getpgid setgrent setreuid
 - getgrgid_r getpgrp setpgid setsid
 - I/O functions
 - pread, pwrite, readv
 - Formatted output
 - snprintf, vsnprintf
- Support for POSIX style UID/GID

20-Jan-2003

Partners Presentation 2003

13



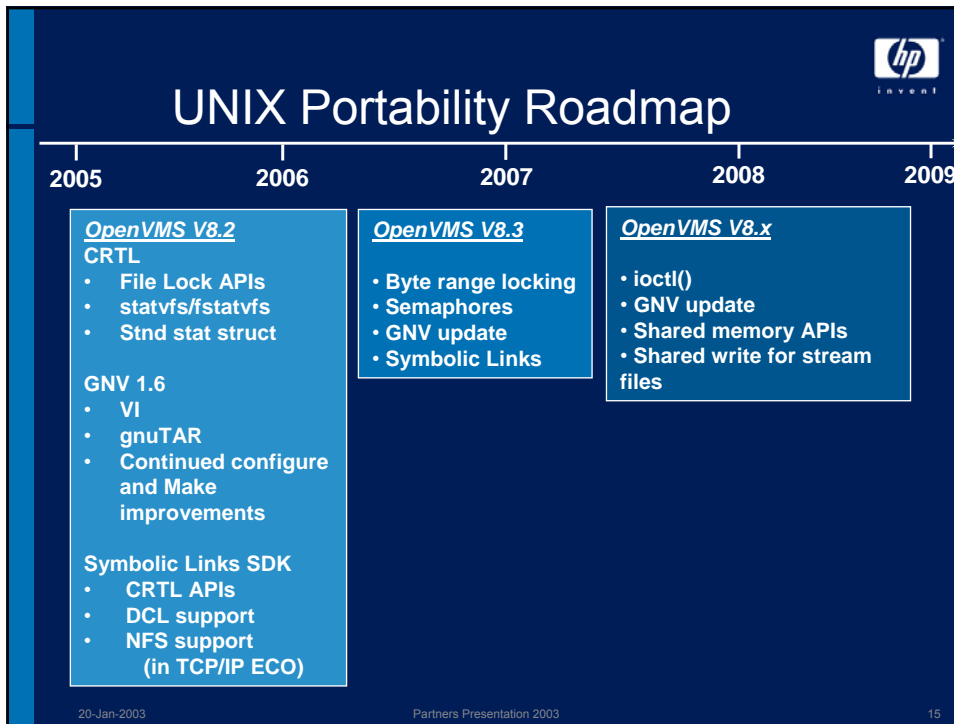
V7.3-2 Enhancements

- TCP/IP Enhancements
 - 64-bit pointer support in: sendmsg, recvmsg, freeaddrinfo, getaddrinfo
 - poll () - input/output multiplexing
 - > 64K data transfers: recv, send, recvfrom, sendto
- UNIX File-Name Enhancements
- Extended DCL command line length
 - Increases DCL command buffer to 4K


20-Jan-2003

Partners Presentation 2003

14




This slide features the HP logo in the top left corner. The text "V8.2" is displayed in a large white font on a blue background. Below this, the text "Planned UNIX Portability Features" is shown in a white box. To the right, there is a large white plus sign followed by the lowercase "hp" logo. At the bottom left, a small copyright notice reads: "© 2005 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice".



V8.2 New Features

- File Locking functions
 - X/Open file locking synchronization in threaded programs
 - flockfile(), ftrylockfile(), funlockfile()
 - clearerr_unlocked(), getc_unlocked(), getchar_unlocked(), feof_unlocked(), ferror_unlocked(), fgetc_unlocked(), fputc_unlocked(), putc_unlocked(), putchar_unlocked()
 - Integrated into all other stdio functions
- File-System Statistics functions
 - statvfs(), fstatvfs()

20-Jan-2003 Partners Presentation 2003 17



V8.2 New Features

- Standard compliant stat structure
 - st_dev and st_rdev fields
 - Declared (char *) on VMS –
 - Declared int in X/Open
 - New feature test macro - `_USE_STD_STAT`
 - Provides the X/Open compliant definition of st_dev and st_rdev
 - Also provide st_blksize and st_blocks
 - X/Open fields that were previously not included
 - Fixes a very frequently seen porting problem

20-Jan-2003 Partners Presentation 2003 18



V8.2 New Features

- `fcntl()` function – added `F_SETFL` and `F_GETFL`
 - Two previously unimplemented function options
 - Set and get file status flags...
- Stream oriented pipes
 - `popen()` creates a CR/LF oriented pipe
 - UNIX pipes do not have such record control
 - `DECC$POPEN_NO_CRLF_REC_ATTR` will cause `popen()` to create a UNIX style pipe
 - More compatible with UNIX expected behaviors

20-Jan-2003

Partners Presentation 2003

19



V8.2 New Features

- `socketpair()`
 - API to create a pair of connected sockets()
 - Requires underlying TCP/IP support
 - TCP/IP Services V5.5 required
- Performance enhancements for `stat`
 - Ongoing evaluation of C RTL performance,
 - Some minor improvements in `stat` – may be significant for file intensive applications

20-Jan-2003

Partners Presentation 2003

20



GNV V8.2 utilities

- GNUtar
 - Archive and back up files
- patch
 - Apply a set of differences to a source file
- du
 - Display amount of disk space used by the contents of a directory
- env, printenv
 - Display environment variables

20-Jan-2003

Partners Presentation 2003

21



GNV V8.2 utilities (continued)

- file
 - Display file type
- su (SuperUser)
 - Become a different user
- Vim (“vi improved”)
 - Invoke a common UNIX text editor
- which
 - Find an executable in the current user’s path
- Improvements to cc, gcc, ld

20-Jan-2003

Partners Presentation 2003

22



hp
invent

Future UP Features

- V8.3
 - Symbolic links & POSIX pathname processing
- V8.3 Candidates
 - Byte-range locking
 - Shared-stream I/O
 - Semaphores

20-Jan-2003 Partners Presentation 2003 24

POSIX pathname processing and symbolic links

What is a symbolic link?



- A symbolic link is a directory entry that associates a name with a text string
- The text string is interpreted as a POSIX pathname when accessed by certain services
- It is implemented on OpenVMS as a file of organization SPECIAL and type SYMBOLIC_LINK
- Symbolic links are also known as “Symlinks”



Example of symbolic link creation

- New DCL qualifier /SYMLINK for CREATE
- Example:

```
$ sho def
SYS$SYSDEVICE:[MCCUSKER.WORK]
$ create/symlink="a/b.txt" link_to_b.txt
$ dir/date link_to_b.txt

Directory SYS$SYSDEVICE:[MCCUSKER.WORK]

LINK_TO_B.TXT;1 -> a/b.txt
                    6-APR-2005 08:13:29.83

Total of 1 file.
$
```

20-Jan-2003

Partners Presentation 2003

27



Example of symbolic link access

- Assume file being referenced does not exist:

```
$ TYPE LINK_TO_B.TXT
%TYPE-W-OPENIN, error opening
SYS$SYSDEVICE:[MCCUSKER.WORK]LINK_TO_B.TXT;1 as input
-RMS-E-ACC, ACP file access failed
-RMS-E-DNF, directory not found
$
$ create/dir [.A]
$
$ TYPE LINK_TO_B.TXT
%TYPE-W-OPENIN, error opening
SYS$SYSDEVICE:[MCCUSKER.WORK]LINK_TO_B.TXT;1 as input
-RMS-E-FNF, file not found
$
```

20-Jan-2003

Partners Presentation 2003

28



- Create the missing file through the link:

```
$ TYPE LINK_TO_B.TXT
%TYPE-W-OPENIN, error opening
SYS$SYSDEVICE:[MCCUSKER.WORK]LINK_TO_B.TXT;1 as in
put
-RMS-E-FNF, file not found
$ create link_to_b.txt
This text is in file [.a]b.txt
Exit
$
```

20-Jan-2003

Partners Presentation 2003

29

Example of symbolic link access (continued)



- Now we can type the file through the link:

```
$ TYPE LINK_TO_B.TXT
This text is in file [.a]b.txt
$ DIR [.a]

Directory SYS$SYSDEVICE:[MCCUSKER.WORK.A]

b.txt;1

Total of 1 file.
$ type [.a]b.txt
This text is in file [.a]b.txt
$
```

- In this example, RMS noticed the input file was a symbolic link, read its contents and interpreted those contents as a POSIX pathname

20-Jan-2003

Partners Presentation 2003

30



RMS support for symbolic links

- `sys$open()`
 - Operates on the target file pointed to by the symbolic link
- `sys$create()`
 - Creates the file pointed to by the symbolic link
- `sys$search()`
 - Returns the DVI and FID of the target file; DID is zero; resultant name is that of the symbolic link and not the target file
- Flag `NAML$V_OPEN_SPECIAL`
 - Causes `sys$open()` and `sys$search()` to not follow the symbolic link

20-Jan-2003

Partners Presentation 2003

31



C RTL support for symbolic links

- Six newly documented APIs:
 - `symlink()` -- create a symbolic link
 - `readlink()` -- read the contents of a symbolic link
 - `unlink()` -- delete a symbolic link
 - `realpath()` -- return a direct pathname from the root
 - `lchown()` -- change the owner of a symbolic link
 - `lstat()` -- return attributes of a symbolic link
- Other APIs that accept pathnames recognize symbolic links

20-Jan-2003

Partners Presentation 2003

32

File system support for symbolic links



- No file system changes until V8.3
- Access subfunction will recognize a symbolic link and convert write access to read access
- Create subfunction will recognize a symbolic link and convert the create to a read access
- For wildcard directory search, lookup will match either a filename ending in “.DIR;1” or a filename which is a symbolic link

20-Jan-2003

Partners Presentation 2003

33

POSIX pathname processing



- To provide a consistent programming environment, developers must be able to use POSIX pathnames through OpenVMS interfaces such as the C RTL and system services
- Other standard POSIX features (system-wide root, mount points, current working directory, version limits) must also be provided
- Rules must be devised to deal with the differences between POSIX pathnames and Open VMS file names

20-Jan-2003

Partners Presentation 2003

34

POSIX pathnames for RMS and DCL



- Issue: The POSIX name-separator '/' character has a different meaning to DCL (as a qualifier indicator)
- Quoting a pathname allows us to pass the pathname through DCL (since quoted strings are already allowed in DCL for DECnet)
- Adding a prefix to the pathname allows RMS to recognize the string as a POSIX pathname
- Format: "^UP^pathname"
- Example: a/b.txt becomes "^UP^a/b.txt"

20-Jan-2003

Partners Presentation 2003

35

DCL POSIX pathname example



```

$ dir [.a]

Directory SYS$SYSDEVICE:[MCCUSKER.WORK.A]

B.C;2

Total of 1 file.
$ cc "^UP^a/b.c"/obj="^UP^a/b.obj"
$
$ link/exe="^UP^a/b.exe" "^UP^a/b.obj"
$
$ dir [.a]

Directory SYS$SYSDEVICE:[MCCUSKER.WORK.A]

B.C;2                b.exe;1                b.obj;1

Total of 3 files.
$

```

20-Jan-2003

Partners Presentation 2003

36



System-wide root

- Available for use with POSIX pathnames
- New ROOT keyword for SET
 - Can specify /SYSTEM for system root or /PROCESS for a process root; /NOPROCESS clears the process root
- Example:

```
$ sho def
DKB0:[TEST]
$ set root /process dkb0:[test]
%SET-I-PSXROOSET, process POSIX root set to
DKB0:[TEST]
$ type "^UP^/a/b.txt"
This is a text file
```

20-Jan-2003

Partners Presentation 2003

37



Mount points

- Allows the crossing of volumes from the root
- New mnt and umnt utilities
- Example:

```
– $ dir dkb100:[newtest]
NEWDIR.DIR;1
Total of 1 file.
$ mnt dkb100:[newtest] /a/mnt
$ dir DKB0:[TEST.A.MNT]
NEWDIR.DIR;1
Total of 1 file
$
```

20-Jan-2003

Partners Presentation 2003

38



Current working directory

- Similar to OpenVMS default directory
- May not be a search list
- Must exist
- Example:
 - \$ SET DEFAULT “^UP^/a/mnt”
 - \$ SHOW DEFAULT
 - DKB100:[NEWDIR]
 - \$

20-Jan-2003

Partners Presentation 2003

39



C RTL and GNV

- The C RTL and GNV will accept pure POSIX pathnames (no need for the ^UP^ quoted format)
- DECC\$POSIX_COMPLIANT_PATHNAMES controls how input pathnames are interpreted
- File versions are treated as on UNIX; only one version of a file will be deleted on a delete operation; no new version of a file is created on a create operation

20-Jan-2003

Partners Presentation 2003

40



File naming

- POSIX allows filenames “a” and “a.” in the same directory
- A file created through GNV and the C RTL that does not have a “.” or that ends in a “.” will have an additional “.” appended to its name to ensure uniqueness
- To allow POSIX filename “a.DIR” and directory “a” to co-exist in the same directory, GNV and the C RTL will append a “.” to a filename ending in “.DIR”

20-Jan-2003

Partners Presentation 2003

41



When can I use this?

- An SDK is planned for V8.2 that contains the images you need to use symbolic links and POSIX pathname processing

20-Jan-2003

Partners Presentation 2003

42



Byte-range locking

- Implemented in the C RTL
 - Fcntl() API
 - F_GETLK, F_SETLK, F_SETLKW options

20-Jan-2003

Partners Presentation 2003

43



Shared-stream I/O

- Shared-write access to stream files
 - Standard UNIX I/O default

20-Jan-2003

Partners Presentation 2003

44



Semaphores

- Implemented via the C RTL
- POSIX semaphores
 - sem_open(), sem_post(), sem_wait(), etc.
- System V semaphores
 - semop(), semctl(), semget()
- Thread-aware
- Node-specific


20-Jan-2003 Partners Presentation 2003 45



Porting Experiences




© 2005 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



Porting Experiences

- Some partners already using UP features (in V7.3-1) to port their applications to OpenVMS
- Recent Experience
 - HP and a partner worked in HP lab to determine level of effort needed to port partner's application
 - Summary of that effort follows on next slides

20-Jan-2003 Partners Presentation 2003 47



Porting Experience

- Application architected to isolate OS specific features
 - An OS interface layer
 - A Network layer
- Source files maintained on partner's Linux system
 - NFS served to OpenVMS system
- Team made extensive use of BASH
 - make and sed used extensively
 - Some minor changes to partner's make files, especially in the area of recursive make
 - Successfully compiled and linked all modules, except missing semaphore routines
 - ar used to populate object libraries
 - Some difficulty with GNV linker, successfully used OpenVMS linker

20-Jan-2003 Partners Presentation 2003 48



Porting Experiences (cont.)

- Successfully passed all tests
 - Developer couldn't believe it - rewrote tests to add verification that it was actually executing properly
- Some things were missing, some hiccups
 - Semaphore support
 - Planned for V7.3-2next
 - poll(), vsnprintf()
 - Both planned for V7.3-2
 - Some trouble with periods in directory names
 - file and lex utilities not yet implemented in bash (planned)
- Overall, a positive experience
 - Partner feels effort to port will be similar to other UNIX® ports

20-Jan-2003

Partners Presentation 2003

49




Porting Experiences #2

- Customer needed a solution for printing barcode labels
 - Simple application, just print the barcode
 - Very expensive to purchase
 - products included more than customer needed
 - Found simple, UNIX Open Source application
 - Downloaded to OpenVMS 7.3-1 with BASH
 - Ran build scripts
 - Everything worked
 - Minimal effort
 - Not all will be this easy, but, this demonstrates the goal

20-Jan-2003

Partners Presentation 2003


50



More Porting Experiences

- From the GNV developers list (July 2003):
 - “GNV is working better and better. I could `./configure` and `make install` the following packages (sometimes with little hacks):
 - `mktemp 1.5`
 - `hostinfo 2.2`
 - `patch 2.5.4`
 - `yacc 1.9.1`
 - `flex 2.5.4`
 - `bison 1.35`”
 - “Recently I gave a try at making a few unix tools I had troubles building in the past, under the latest GNV bash, and got surprisingly further along, than in the past”

20-Jan-2003 Partners Presentation 2003 51



GNV Tip

- Define `DECC$PIPE_BUFFER_SIZE 65000` to maximize pipe capabilities
 - Most configure scripts will need this set.
- New feature/parameter `DECC$PIPE_BUFFER_QUOTA`
 - Exploits VMS 7.3-1 change that increases mailbox buffer quotas (`$crembx:bufquo`)
 - Be careful – given enough `BYTLM`, processes can quickly eat up virtual memory

20-Jan-2003 Partners Presentation 2003 52



Contacts

- Engineering Lead/Strategic Direction
Gaitan.Dantoni@hp.com
- OpenVMS UNIX Portability Tech Lead:
Paul.Cerqua@hp.com
- OpenVMS UNIX Portability Business Manager:
Leo.Demers@hp.com
- Brad McCusker
Brad.mccusker@hp.com
- <http://h71000.www7.hp.com/portability/index.html>

20-Jan-2003

Partners Presentation 2003

53

