Porting Open Source to OpenVMS

Brad McCusker
OpenVMS Engineering

HP WORLD 2003
Solutions and Technology Conference & Expo

Topics

- UNIX Application Portability Initiative - Overview
- Future Plans
  - OpenVMS V7.3-2
  - OpenVMS V8.2
  - Future Releases
- Porting Experiences

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

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

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)

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

Rollout…

- Started already with VMS V7.3-1…
  - Delivered first set of UNIX “enabling technologies”
    - C RTL
    - GNV/BASH (Commands & Utilities)
    - File system improvements

OpenVMS 7.3-1 Enhancements

- File system
  - Mixed case file names, case sensitive compares
  - Time of last file access
  - Hard link improvements
  - Root directory support

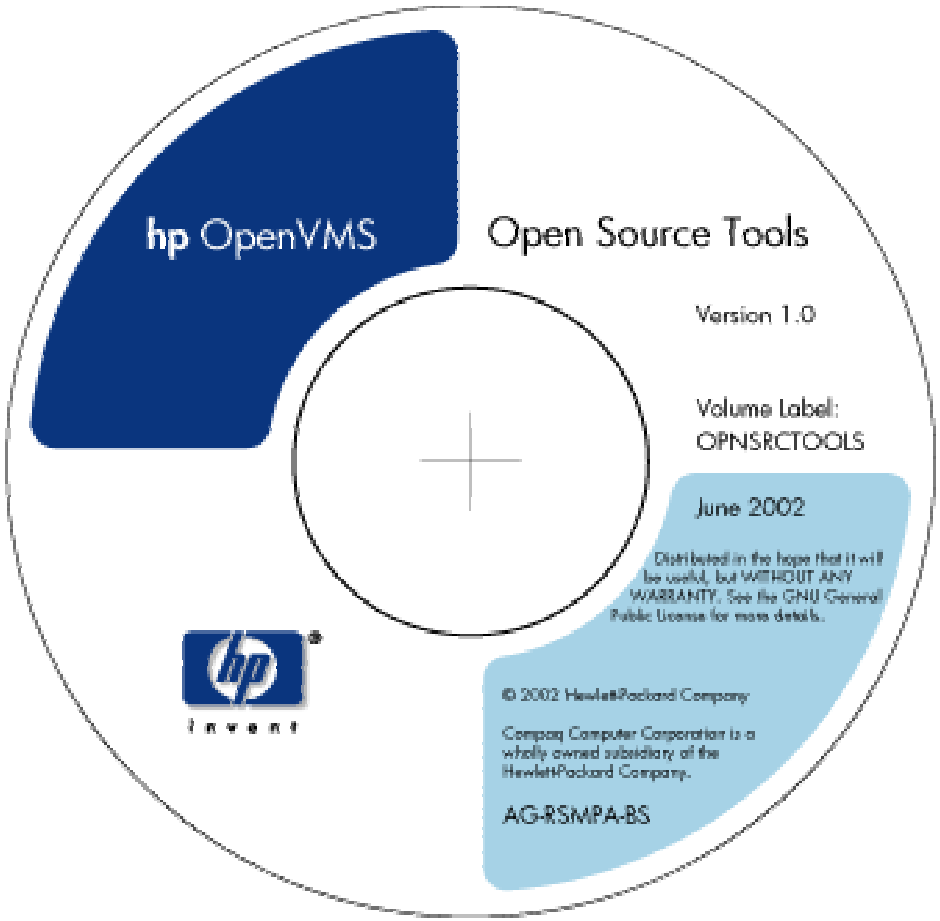
- C Run Time Library
  - New UNIX APIs
  - Improved UNIX filename support
  - API for controlling feature switches

OpenVMS 7.3-1 – 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

  - Updated version included with OpenVMS and at: http://h71000.www7.hp.com/opensource/opensource.html
  - Updates include:
    - ODS-5 file system support
    - Additional utilities ported and included
    - Packaged as a HP-branded PCSI kit

Open Source Tools CD

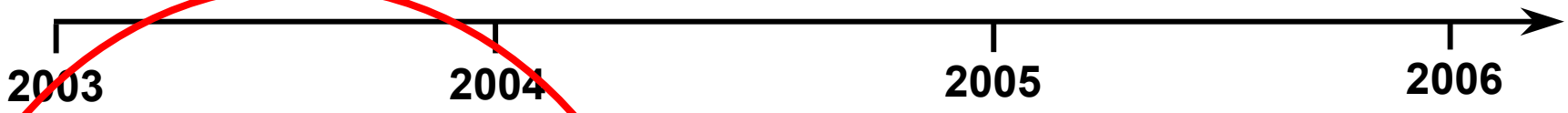


- Open Source Tools CD
- Ships with OpenVMS V7.3-1
- Includes GNV
- Other contents:
  - Stunnel
  - VMSTAR
  - ZIP
  - SSL 0.9.6b Sources
  - CDRECORD
  - OpenVMS Migration Software

What does the future hold?

HP WORLD 2003
Solutions and Technology Conference & Expo

UNIX Portability Roadmap



OpenVMS Alpha V7.3-2 (Opal)

- CRTL
  - •User database functions
  - •Unix style signal functions
  - •UID/GID
  - •poll()
  - •File I/O
- •GNV 1.4
  - •Bbzip2
  - •Gawk
  - •Man
  - •Improved complier interfaces (cc, gcc)

OpenVMS V8.2 (Topaz)

- CRTL
  - •Binary Tree
  - •Asynch I/O interfaces
  - •Symbolic links
  - •glob()
  - •flock()
- •GNV
  - • VI
  - • Configure and make improvements

OpenVMS UP future investigations

- • Full function select()
- • fork ()
- •Semaphores
- • Byte range locking
- • Shared memory APIs
- • Unix 98 compliance

OpenVMS I64 V8.1

- • CRTL
- • GNV

Note: Investigations are provided solely to inform what is being considered and should not be used as a deliverable commitment.

V7.3-2 C RTL Contents (tentative)

- Support for POSIX style UID/GID
  - Requires 32 bit UID/GID data structures
    - Define _ _USE_LONG_GID_T macro to compile an application for 32-bit UID/GID support
  - DECC$POSIX_STYLE_UID controls whether UIDs/GIDs are POSIX style or derived from the process UIC.
  - APIs affected:
    - getegid, geteuid, getgid, getuid, setgid, setuid and others

V7.3-2 C RTL Contents (tentative)

- 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

V7.3-2 C RTL Contents (tentative)

- New APIs – glob(), globfree()
  - Pattern matching APIs
  - Extended to allow for VMS style behavior
    - Controlled by feature switch – default is VMS behavior
    - Use ‘*’, ‘&’, ‘…’ for wildcard, not ‘?’
    - No pattern matching
  - (Feature switch name – TBD – check the docs)

V7.3-2 C RTL Contents (tentative)

- 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 interface extension
  - pread, pwrite readv
- Formatted output
  - snprintf, vsnprintf

V7.3-2 C RTL Contents (tentative)

- TCP/IP related enhancements
  - poll () - input/output multiplexing
    - Limited to sockets only
  - 64-bit pointer support in: sendmsg, recvmsg, freeaddrinfo, getaddrinfo
    - Previously, these functions had only a 32-bit interface.
  - > 64K data transfers: recv, send, recvfrom, sendto,
    - Previously limited to 64K bytes.
    - Support is latent

      > 64K requires support in underlying TCP/IP stack

      Not in TCP/IP Services V5.4

UNIX File-Name Translation Enhancements
Performance Enhancements

- New cache for logical name translation
  - DECC$ENABLE_TO_VMS_LOGNAME_CACHE
    - Speeds translation of logical names in UNIX file name translation.

      0 – Cache disables (default)

      1 – Enable cache with 1 second entry life

      2 – Enable cache with 2 second entry life

      Etc

      -1 – Enable cache with no entry expiration time

UNIX File-Name Translation Enhancements Performance Enhancements (2)

- DECC$EFS_NO_DOTS_IN_DIRNAME
  - NAME.EXT can be:
    - File – []NAME.EXT
    - Directory [.NAME^.EXT]
  - Checking to see if [.name^.ext] adds overhead.
  - Enable DECC$EFS_NO_DOTS_IN_DIRNAME to suppress interpretation of a filename with dots as a directory.

UNIX File-Name Translation Enhancements

- DECC$NO_ROOTED_SEARCH_LISTS
  - Controls how the to_vms function resolves search-list logicals.
  - ENABLE – Assume a search list logical is all non-rooted logicals
  - V7.3 behavior - /dev translates to dev:
  - V7.3-1behavior - /dev translates to:
    - dev is rooted logical: dev:[000000]
    - dev is non-rooted logical: dev:
    - dev is search list: look at first element and translate as above
  - .

UNIX File-Name Translation Enhancements & bug Fixes (cont)

- DECC$NO_ROOTED_SEARCH_LISTS (cont)
  - If dev is a search list of mixture of rooted and non-rooted logicals – translation can break and may not match legacy.
  - By assuming non rooted search lists, DECC$NO_ROOTED_SEARCH_LISTS restores legacy behaviors
- Angle brackets now correctly recognized in UNIX name translation.
  - Previously, we weren't interpreting '<' & '>' as directory delimeters

V7.3-2 C RTL Contents
New Feature Switches

- DECC$EXEC_FILEATTR_INHERITANCE
  - Feature logical modified to allow greater choice in inheritance of file access modes.

    = 1 child inherits file positions for all file access modes except append.

    = 2 child inherits file positions for all file access modes including append.

    = 0 (disabled) child process does not inherit the file position.

V7.3-2 C RTL Contents
New Feature Switches (2)

- DECCUSE_JPI_CREATOR
  - Affect processing of getppid()
  - ENABLE
    - Use $GETJPI/JPI$_CREATOR to determines parent process ID
    - UNIX compliant behavior
  - DISABLE
    - Use $GETJPI/JPI$_OWNER to determines parent process ID
    - Traditional VMS behavior

V7.3-2 C RTL Contents
New Feature Switches (3)

- DECC$ALLOW_REMOVE_OPEN_FILES
  - remove() of an open file typically fails
  - Standard compliance dictates that the operation succeed
  - Enable this feature logical to enable the standard compliant behavior.

- DECC$ALLOW_UNPRIVILEGED_NICE
  - Controls legacy vs. standard complaint behavior
  - ENABLE
    - Exhibits legacy behavior
      No privilege check on calling process
      Set to value > MAX_PRIORITY sets to base priority
  - DISABLE
    - Conforms to the X/Open standard
      Check privilege of the calling process (ALTPRI is needed)
      Set to value > a MAX_PRIORITY, sets to MAX_PRIORITY

V7.3-2 C RTL Contents
New Feature Switches (4)

- New switch – DECC$RENAME_ALLOW_DIR
  - rename() to directory is non-UNIX standard
    - But, it is VMS standard behavior
    - Example:

      `rename (file.ext,logname)`

      Where:

      `logname = [dir.subdir]`

      Results in:

      `[dir]subdir.ext`

      This happens because logname gets translated to a file because rename to a directory is not allowed
  - This switch restores the VMS behavior

    `rename (file.ext,logname)` → **`[dir.subdir]file.ext`**

A word about rename()

- DECC$RENAME_NO_INHERIT should have been called RENAME_UNIX_COMPATIBLE
  - DECC$RENAME_NO_INHERIT causes UNIX compliant behaviors to be enforced –

- When DECC$RENAME_NO_INHERIT is enabled, DECC$RENAME_ALLOW_DIR is ignored.

V7.3-2 C RTL Contents (tentative)

- Extended command line length
  - VMS 7.3-2 increases DCL command buffer to 4K
  - Corresponding changes in C RTL to support larger command lines

V7.3-2 C RTL Contents
New API

- decc$set_child_default_dir
  - Typically, vfork/exec child processes inherit default directory from the parent.
  - decc$set_child_default(default_dir)
    - Subsequent calls to vfork/exec, child processes created with default directories set to default_dir

7.3-2 CRTL Contents
Enhanced access()

- access() enhanced to also check ACLs
  - DECC$ACL_ACCESS_CHECK
  - Uses $checkpro system service
  - Eventually need to add similar capability to stat() and other APIs – not done yet though

GNV Release for V7.3-2

- New utilities
  - bbzip2
  - gawk
  - man
  - GNU TAR
- Working towards getting “configure” to work for any arbitrary Open Source package.
- IPF Port complete – Negligible effort!
  - Available in E8.1 time frame
  - Native compiler support may lag behind

UNIX Portability Roadmap

OpenVMS Alpha V7.3-2 (Opal)

- CRTL
  - User database functions
  - Unix style signal functions
  - UID/GID
  - poll()
  - File I/O
- GNV 1.4
  - Bbzip2
  - Gawk
  - Man
  - Improved complier interfaces (cc, gcc)

OpenVMS V8.2 (Topaz)

- CRTL
  - Binary Tree
  - Asynch I/O interfaces
  - Symbolic links
  - glob()
  - flock()
- GNV
  - VI
  - Configure and make improvements

OpenVMS I64 V8.1

- CRTL
- GNV

OpenVMS UP future investigations

- Full function select()
- fork ()
- Semaphores
- Byte range locking
- Shared memory APIs
- Unix 98 compliance

Note: Investigations are provided solely to inform what is being considered and should not be used as a deliverable commitment.

V8.2 Release Contents (tentative)

- Semaphores
  - semctl, semget, semop, sem_close, sem_destroy, sem_getvalue, sem_init, sem_open, sem_post, sem_trywait, sem_unlink, sem_wait
- Symbolic links
  - symlink, lstat, lchown, readlink
- Locking
  - flockfile, ftrylockfile, funlockfile
- Asynch I/O interfaces
  - aio_cancel, aio_error, aio_fsync, aio_read, aio_return, aio_suspend, aio_write

V8.2 Release Contents (tentative)

- Binary Tree
  - tdelete, tfind, tsearch, twalk
- Others still to be determined

UNIX Portability Roadmap

OpenVMS Alpha V7.3-2 (Opal)

- CRTL
  - •User database functions
  - •Unix style signal functions
  - •UID/GID
  - •poll()
  - •File I/O
- •GNV 1.4
  - •Bbzip2
  - •Gawk
  - •Man
  - •Improved complier interfaces (cc, gcc)

OpenVMS V8.2 (Topaz)

- •CRTL
  - •Binary Tree
  - •Asynch I/O interfaces
  - •Symbolic links
  - •glob()
  - •flock()
- •GNV
  - • VI
  - • Configure and make improvements

OpenVMS UP future investigations

- • Full function select()
- • fork ()
- •Semaphores
- • Byte range locking
- • Shared memory APIs
- • Unix 98 compliance

OpenVMS I64 V8.1

- • CRTL
- • GNV

Note: Investigations are provided solely to inform what is being considered and should not be used as a deliverable commitment.

Future Releases Contents

- Future releases plan to add APIs to achieve industry standard compliance (UNIX98? POSIX? LINUX?…)
- The following areas will be covered:
  - fork()
  - File System
  - Resource management
  - Byte Range Locking
  - Messaging functions
  - Schedulers functions
  - TCP
  - Other UNIX tools
  - Increase Command Line length
  - UNIX style shareable images
  - Shared memory

Future Releases Contents

- Fork()
  - New system service $CLONE_PROCESS
  - IR complete, Functional Spec under review
  - Post Itanium time frame
- UNIX I/O
  - aka “forkable-IO”
  - Goes hand-in-hand with fork()
- File System
  - select() – pause until specified activity is detected
    - Full featured (sockets, files, pipes, etc)
  - fallocate
  - fstatvfs - get file system information (maybe V7.3-2)
  - statvfs - get file system information (maybe V7.3-2)
  - ulimit - set or report file size limit

Future Releases Contents

- Resource management
  - getrlimit - get maximum resource settings
  - madvise
  - setrlimit - set maximum resource consumption
  - getrusage - get information about resource utilization
- Messaging functions
  - msgctl - message control operations
  - msgget - get the message queue identifier
  - msgrcv - message receive operation
  - msgsnd - message send operation

Future Releases Contents

- Schedulers functions
  - sched_get_priority_max, sched_get_priority_min - get priority limits(REALTIME)
  - sched_getparam - get scheduling parameters
  - sched_getscheduler - get scheduling policy
  - sched_rr_get_interval - get execution time limits
  - sched_yield - yield processor
- TCP
  - socketpair - create a pair of connected sockets
  - ioctl - control a STREAMS device

Future Releases Contnets

- Miscellaneous
  - crypt - string encoding function(CRYPT)
  - lio_listio - list directed I/O (REALTIME)
  - realpath - resolve a pathname (into an absolute path)

Porting Experiences

HP WORLD 2003
Solutions and Technology Conference & Expo

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

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

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

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

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”

GNV Tip

- define DECC$PIPE_BUFFER_SIZE 65535 to maximize pipe capabilities
- New feature/parameter DECC$PIPE_BUFFER_QUOTA
  - Exploits VMS 7.3-1change that increases mailbox buffer quotas ($crembx:bufquo)
  - Be careful – given enough BYTLM, processes can quickly eat up virtual memory

Contacts

- OpenVMS C RTL Project Leader:
  - Brad.McCusker@hp.com
- OpenVMS UNIX Portability Program Manager:
  - Vittorio.Mezzano@hp.com
- Coming soon: UP Web Site
  -


hp
invent