Porting OpenVMS

September 29, 2014

vms Software

Disclaimer

This presentation contains forward looking statements and is provided solely for your convenience. While this contains our current thinking, all information is subject to change without notice. Information will be updated as planning progresses.

Porting ‘Degree of Difficulty’ Perspective

Alpha

- Expertise: Clueless
- Environment: Familiar
- Little thought to the future

Itanium

- Expertise: Experienced
- Environment: Unfamiliar
- More thought to the future

X86

- Expertise: Very Experienced
- Environment: Predictably Unfamiliar
- Even more thought to the future

Porting Play Book

Chapter 1 – Creating and Executing Images

- Calling Standard, Register Mapping
- Compilers, Assembler
- LIBRARIAN, LINKER
- INSTALL, Image Activator
- Tools: SDA, DEBUG/XDELTA, ANALYZE IMAGE, ANALYZE OBJECT

Chapter 2 – Architecture-Specific Needs (a.k.a. The 5%)

- Booting
- Interrupts, Exceptions
- Memory Management: protection types, access modes, address space, etc.
- Atomic Instructions
- Floating Point
- Special needs for code in assembler (e.g. QUEUE instruction emulation)

Chapter 3 – Compiling and Linking Everything Else (a.k.a. The 95%)

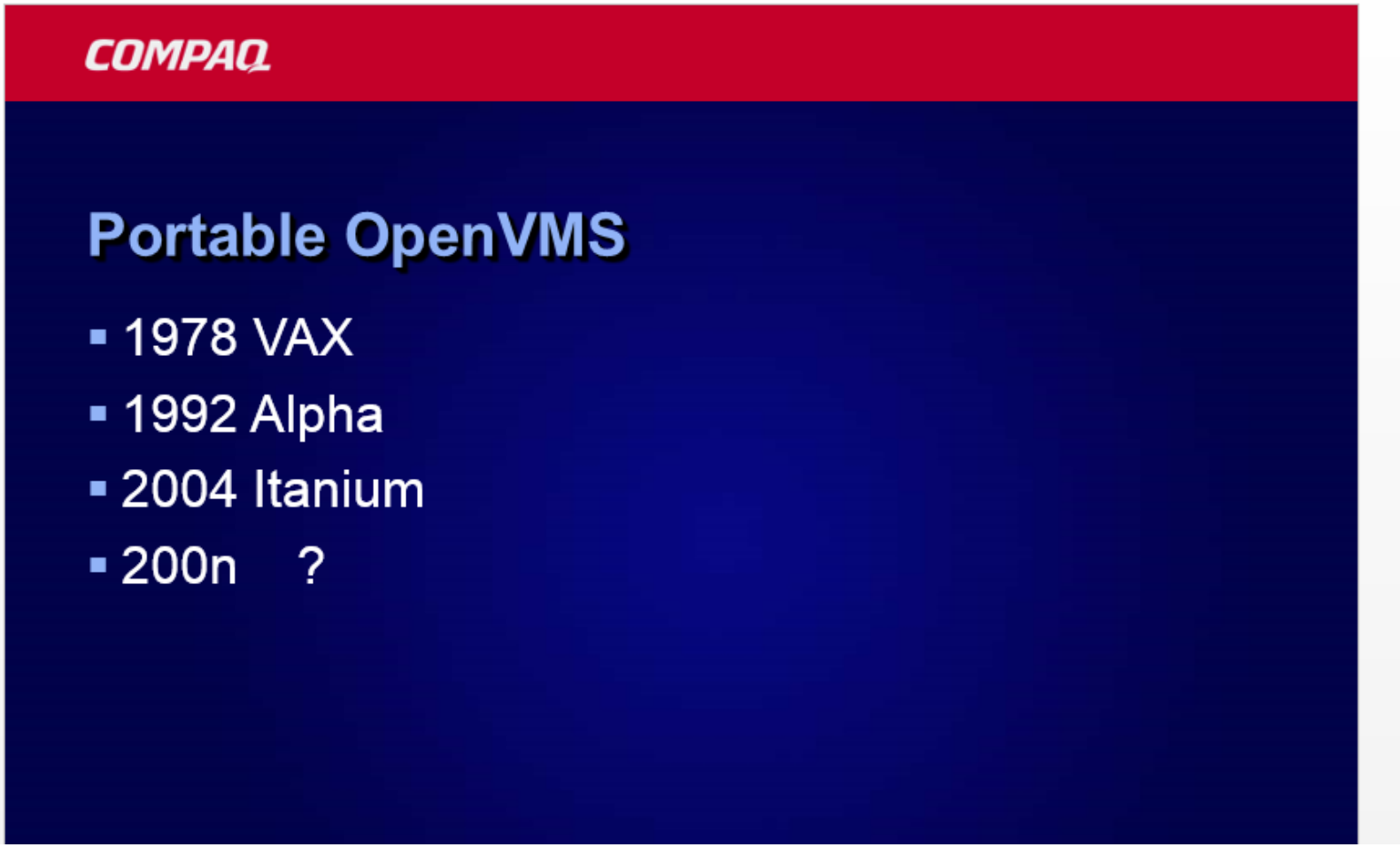
Sequence of Significant Events

1. Boot Contest
2. Full Build
3. Compiler Testing (compilers not needed to boot)
4. Layered Product Testing
5. Build with Native Tools
6. Partner Release
7. General Release for HP DLxxx

Beyond the Initial Release (not necessarily in this order)

- More x86 Platforms
- VMS as a VM guest on one or more x86-based hosts
- Other architectures

A Previous (and very bad) Prediction




vms Software