OpenVMS
(Virtual Memory System)

André Wendt

June 20, 2006
Outline

- History
- Definitions
- System Concepts
  - Process lifecycle
  - Scheduling and memory management
  - Queues
  - DCL
  - File subsystem
  - System monitoring
- Practical examples
- References
OpenVMS History

- 1960s: Digital's PDP series was a successful line of computers
- Early 1970s: PDP arch (16-bit) became too limited
- 1977: First VAX/VMS systems with 32-bit (while PDP-11 was still shipped)
- Performance and capacity of VAX arch increased tremendously
- 1980s: MicroVAX running MicroVMS (later obsoleted)
- 1991: VAX/VMS rename into OpenVMS
- 1992: DEC introduces Alpha AXP (64-bit) to replace VAX
- 2001: Compaq begins porting to IA-64
Definitions

☐ Process:
  - independent, schedulable task under the OS
  - contains running programs, open files, identity, access rights, ...

☐ Types of processes:
  - interactive
    - directly associated with a user
    - created upon login
  - detached
    - can continue after parent terminates
  - batch
    - executes a submitted command procedure on a user's behalf
  - subprocess:
    - child process that performs for and depends on parent
 Definitions (cont'd)

☐ Job:
   - collections of processes that are in a parent-child-relationship

☐ Thread:
   - schedulable task under a process

☐ Queues:
   - lists of jobs scheduled for execution
   - print, batch, and server queues are distinguished
   - types: generic, execution

☐ Image:
   - user program executing as part of the process

☐ System services, commands:
   - denoted as $SCHDWK, $SHOW
Concepts: Process lifecycle

- LOGIN
  - process creation
- DCL inclusion
  - image
    - activation
    - execution
    - rundown
- LOGOUT
  - process deletion

- DCL: Digital Command Language, the command-line interpreter
- not all DCL commands require an image
- image execution can be interrupted
- images are activated and rundown
- process remains intact throughout the terminal session
- separation of the image from the process is unique to OpenVMS
- create a subprocess using $SPAWN
- activate an image using $RUN (otherwise, image must be $INSTALLled)
Concepts: Scheduling – Process states, priorities

- **Process states:**
  - CUR: Executing
  - COM: Ready
  - HIB: Hibernate

- **Process priorities:**
  - Range from [0..31]
  - Real-time processes: [16..31]
  - Interactive processes: [4..9]
  - Batch: base 3
  - Remaining priorities are assigned at system manager's discretion
  - Priority of real-time processes are not changed, they preempt interactive ones
Priority boosts:

- disk I/O complete: +2
- $SCHDWK$ wakeup: +3
- Terminal output complete: +4
- Terminal input complete: +6
- Process created: +6
OpenVMS Memory management is subdivided into page fault handlers, page management, **SWAPPER**, memory status database

- page size ranges from 512 (VAX) to 8192 bytes (Alpha)
- page of the image is copied from disk to memory on demand
- an attempt to access a page not in memory results in a *page fault*
  - hard page fault: required page is on disk
  - soft page fault: required page is in memory, but not in working set

- OS keeps free and modified lists to maintain the remainder of memory not in the working set
- page fault algorithm: FIFO
Concepts: Memory management – Paging (cont'd)

- Page Table Entry (OpenVMS):

<table>
<thead>
<tr>
<th>V</th>
<th>Protection</th>
<th>M</th>
<th>reserved</th>
<th>Frame number</th>
</tr>
</thead>
</table>

- Page Table Entry (Windows 2000):

<table>
<thead>
<tr>
<th>Frame number</th>
<th>reserved</th>
<th>D</th>
<th>A</th>
<th>other</th>
<th>V</th>
</tr>
</thead>
</table>
Concepts: Memory management – Swapping

- Outswapping: removal of inactive processes from memory
- Swapping is done by process (SWAPPER) rather than by procedures
  - under scheduler's control
  - always in HIB state at priority 16 (real-time)
  - periodically (once per sec) checked if swapper's state should be changed to COM

- Basic algorithm:
  - if modified list is too large, copy it to paging file on disk
  - hard page fault: cluster of faulted page+neighbors is requested
  - if free space is too small, reduce all working sets (by moving a portion to the free/modified lists)
  - last resort: eliminate inactive processes by storing whole processes (entire working sets) in swapping file
Concepts: Memory model
Concepts: Queues

- **Queues by purpose:**
  - print – sending output to printers
  - batch – scheduling the execution of command procedures
  - server – submitting files to be processed by a given program run as detached process

- **Types of queues:**
  - generic: route jobs to next available execution queue
  - execution queues

- **Q. entries are assigned priorities, can be processed by seniority**

- **Server queues process any type of file (e.g. e-mail)**
Concepts: Digital Command Language (DCL)

- **Format**: command = verb { qualifier, parameters, keywords }
- **Verb**: What to do?
- **Qualifier**: How to do it? (preceded by a slash)
- **Parameter**: What objects?
- **Keyword**: Non-arbitrary parameter

**Example**: $PRINT /COPIES=2 MAIN.C,MAIN.H

- Positional qualifier: $PRINT MAIN.C,MAIN.H /COPIES=2
- Command (or global) qualifier:
  - $PRINT /QUEUE=P2 MAIN.C+MAIN.H
  - $PRINT MAIN.C+MAIN.H /QUEUE=P2

- Parameter vs. keyword:
  - $DELETE MAIN.C,MAIN.H vs. $SHOW TIME
Incomplete commands are prompted:

- $RENAME
- _FROM_: DRAFT4.TXT
- _TO_: FINAL.TXT

Abbreviations:

- $DIR and $DIRECTORY are synonyms
- user-defined: $DEL*ETE :== DELETE/CONFIRM
Concepts: File Subsystem

- Files-11 originated in RSX on PDP-11
- RSX files are horizontal (same level, no subdirectories)
- OpenVMS: three parts of a file
  - directory, points to the
  - header, locates the
  - data
- header located in file $DISK1:[000000]INDEXF.SYS
- additional features:
  - access control lists
  - file versioning
  - network access
Concepts: System Monitoring (1 of 4)

$SHOW USER[/FULL]$
$SHOW PROCESS/CONTINUOUS$

<table>
<thead>
<tr>
<th>State</th>
<th>CUR</th>
<th>Working set</th>
<th>441</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cur/base priority</td>
<td>8/4</td>
<td>Virtual pages</td>
<td>5667</td>
</tr>
<tr>
<td>Current PC</td>
<td>FFFEE98E</td>
<td>CPU time</td>
<td>000:00:00:05.18</td>
</tr>
<tr>
<td>Current PSL</td>
<td>03C00000</td>
<td>Direct I/O</td>
<td>223</td>
</tr>
<tr>
<td>Current user SP</td>
<td>7FDB991C</td>
<td>Buffered I/O</td>
<td>2752</td>
</tr>
<tr>
<td>PID</td>
<td>2020932A</td>
<td>Page faults</td>
<td>44458</td>
</tr>
<tr>
<td>UIC</td>
<td>[WENDT]</td>
<td>Event flags</td>
<td>E03D00007 C0000000</td>
</tr>
</tbody>
</table>

$1$DKAO:[SYSO.SYSCOMMON.][SYSEXESHOE.XE]
Concepts: System Monitoring (3 of 4)

$\text{MONITOR SYSTEM}$

Node: FAFNER  
Statistic: CURRENT

OpenVMS Monitor Utility  
18-JUN-2006 13:47:59
SYSTEM STATISTICS

Process States
- LEF: 38  LEFO: 0
- HIB: 33  HIBO: 0
- COM: 0  COMO: 0
- PPW: 0  Other: 1
- MWAIT: 0
Total: 72

CPU Busy (14)
- CPU: 0  200
Cur Top: BATCH_799 (8)

Page Fault Rate (0)
- MEMORY: 0  100  0
Cur Top: (0)

Direct I/O Rate (1)
- I/O: 0  60  0
Cur Top: BATCH_799 (1)

Free List Size (481460)
- 873K

Modified List Size (41351)
- 142K

Buffered I/O Rate (481)
- 150
Cur Top: BATCH_799 (450)
Concepts: System Monitoring (4 of 4)

$MONITOR STATE

OpenVMS Monitor Utility
PROCESS STATES

on node FAFNER
19-JUN-2006 16:16:42.40

Collided Page Wait
Mutex & Misc Resource Wait
Common Event Flag Wait
Page Fault Wait
Local Event Flag Wait
Local Evt Flg (Outswapped)

Hibernate
Hibernate (Outswapped)
Suspended
Suspended (Outswapped)
Free Page Wait
Compute
Compute (Outswapped)
Current Process
Practical examples

- timer example
- submitting a job
- submitting a job that re-submits itself
References