# 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

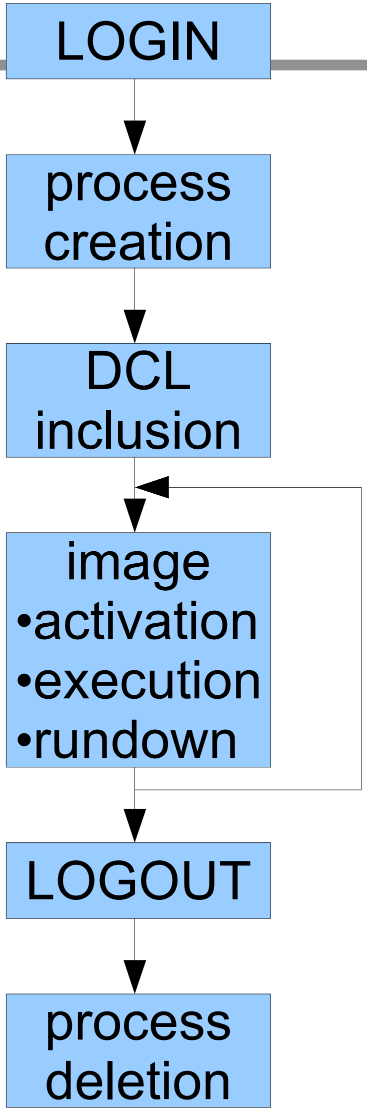
# Definitions

- □ 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



□ 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

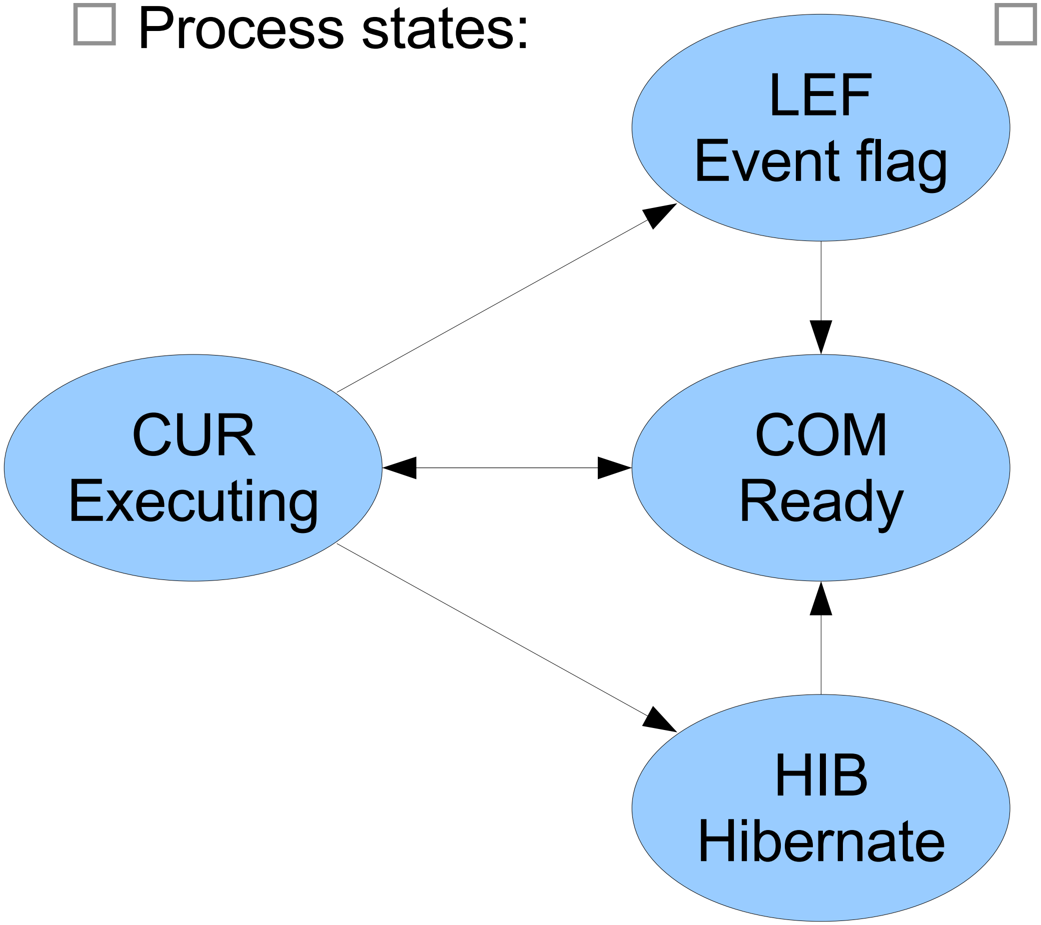
□ separation of the image from the process is unique to OpenVMS

□ create a subprocess using `$SPAWN`

□ activate an image using `$RUN` (otherwise, image must be `$INSTALL`ed)

# Concepts: Scheduling – Process states, priorities

□ Process states:



□ 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

# Concepts: Scheduling – Priority boosts

□ Priority boosts:

- disk I/O complete: +2
- `$SCHDWK` wakeup: +3
- Terminal output complete: +4
- Terminal input complete: +6
- Process created: +6

# Concepts: Memory management – Paging

- □ 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):

| | | | | | |
|---|---|---|---|---|---|
| V | Protection | M | | reserved | Frame number |

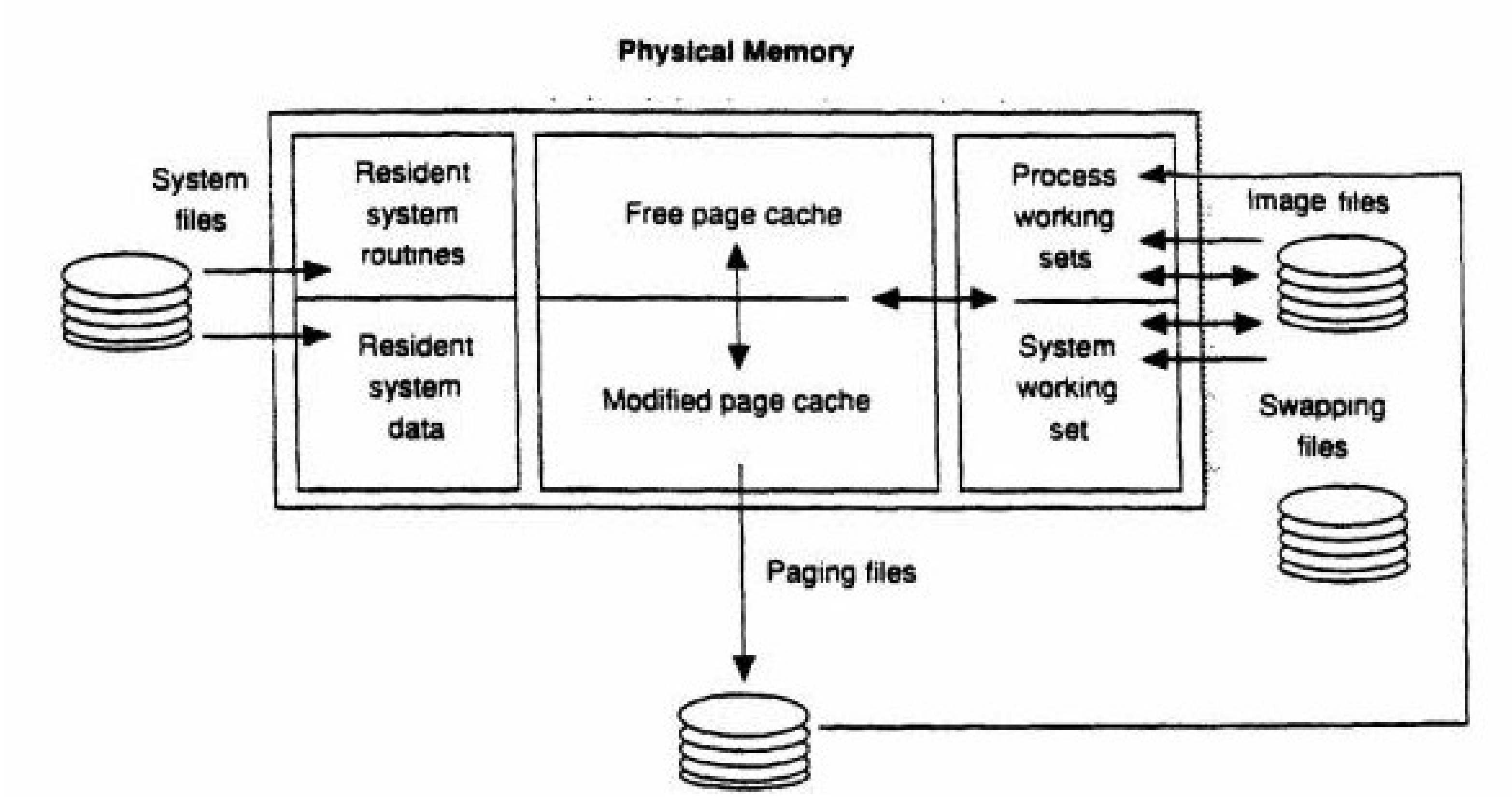
□ Page Table Entry (Windows 2000):

| | | | | | |
|---|---|---|---|---|---|
| Frame number | reserved | D | A | other | V |

# 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`

# Concepts: DCL (cont'd)

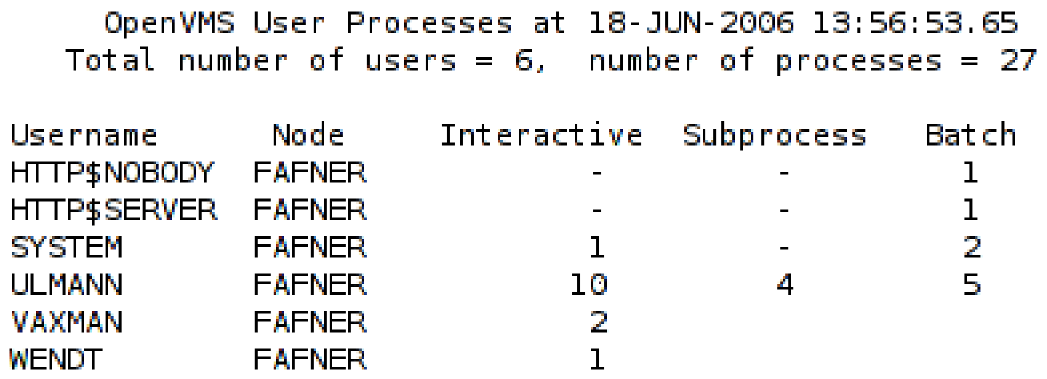
- Incomplete commands are prompted:
  - `$RENAME`
  - `_FROM: DRAFT4.TXT`
  - `_TO: FINAL.TXT`
- Abbreviations:
  - `$DIR` and `$DIRECTORY` are synonyms
  - user-defined: `$DEL*ETE :== DELETE/CONFIRM`

# 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)

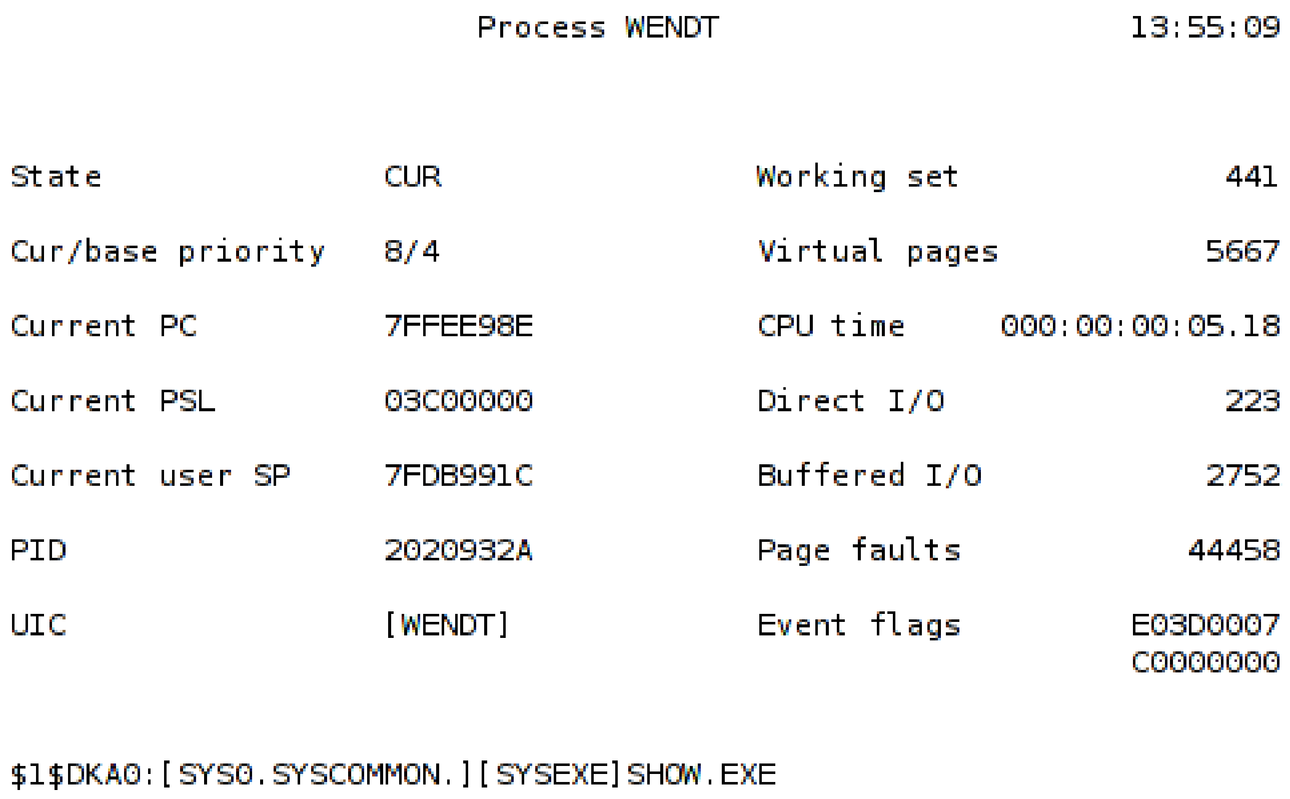
□ $SHOW USER[/FULL]



```
      OpenVMS User Processes at 18-JUN-2006 13:56:53.65
    Total number of users = 6,  number of processes = 27

 Username      Node      Interactive  Subprocess   Batch
 HTTP$NOBODY   FAFNER              -           -       1
 HTTP$SERVER   FAFNER              -           -       1
 SYSTEM        FAFNER              1           -       2
 ULMANN        FAFNER             10           4       5
 VAXMAN        FAFNER              2
 WENDT         FAFNER              1
```

# Concepts: System Monitoring (2 of 4)

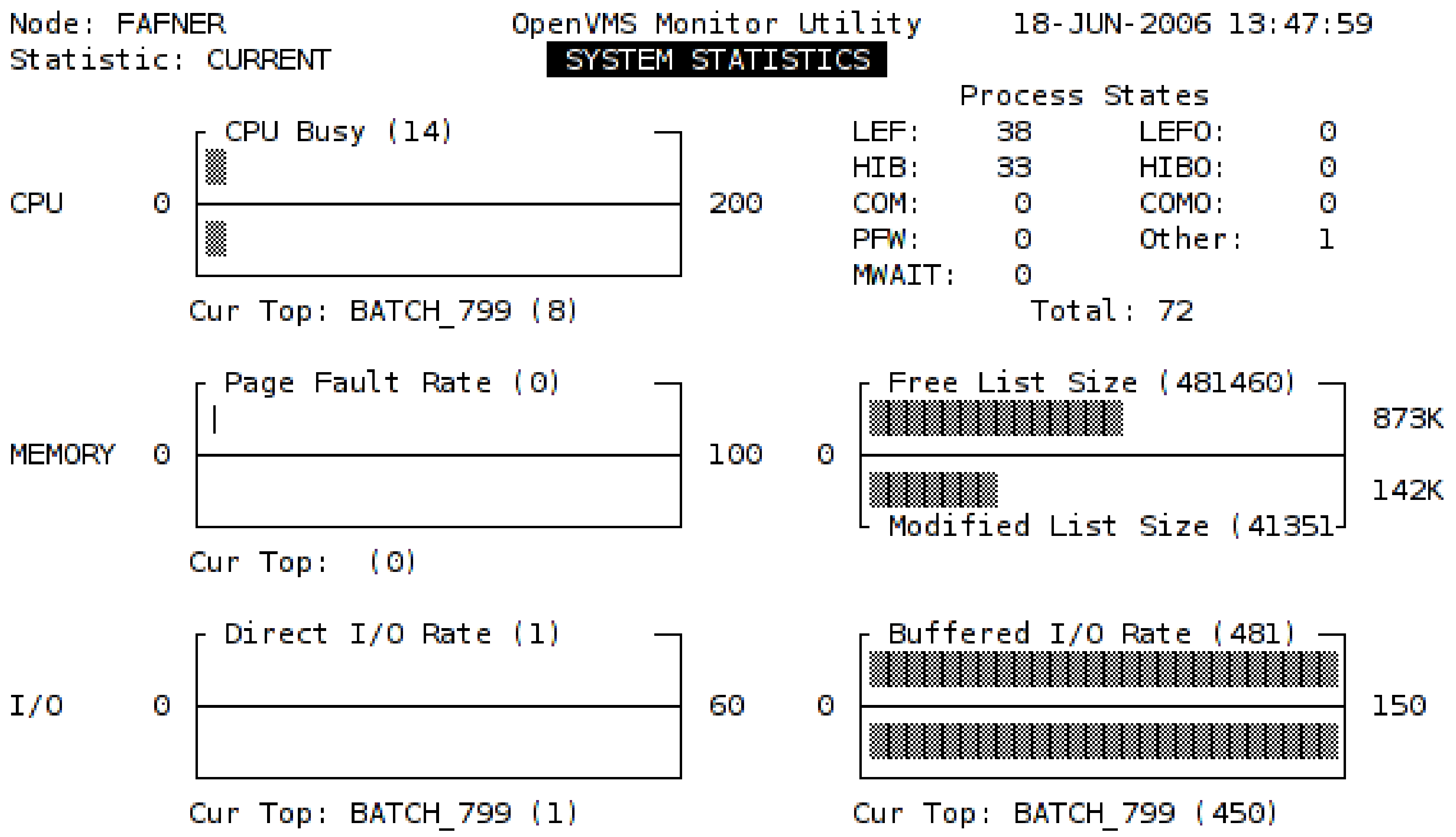
- $SHOW PROCESS/CONTINUOUS



```
                         Process WENDT                       13:55:09


State               CUR                Working set                441

Cur/base priority   8/4                Virtual pages             5667

Current PC          7FFEE98E           CPU time      000:00:00:05.18

Current PSL         03C00000           Direct I/O                 223

Current user SP     7FDB991C           Buffered I/O              2752

PID                 2020932A           Page faults              44458

UIC                 [WENDT]            Event flags           E03D0007
                                                             C0000000


$1$DKA0:[SYS0.SYSCOMMON.][SYSEXE]SHOW.EXE
```

# Concepts: System Monitoring (3 of 4)

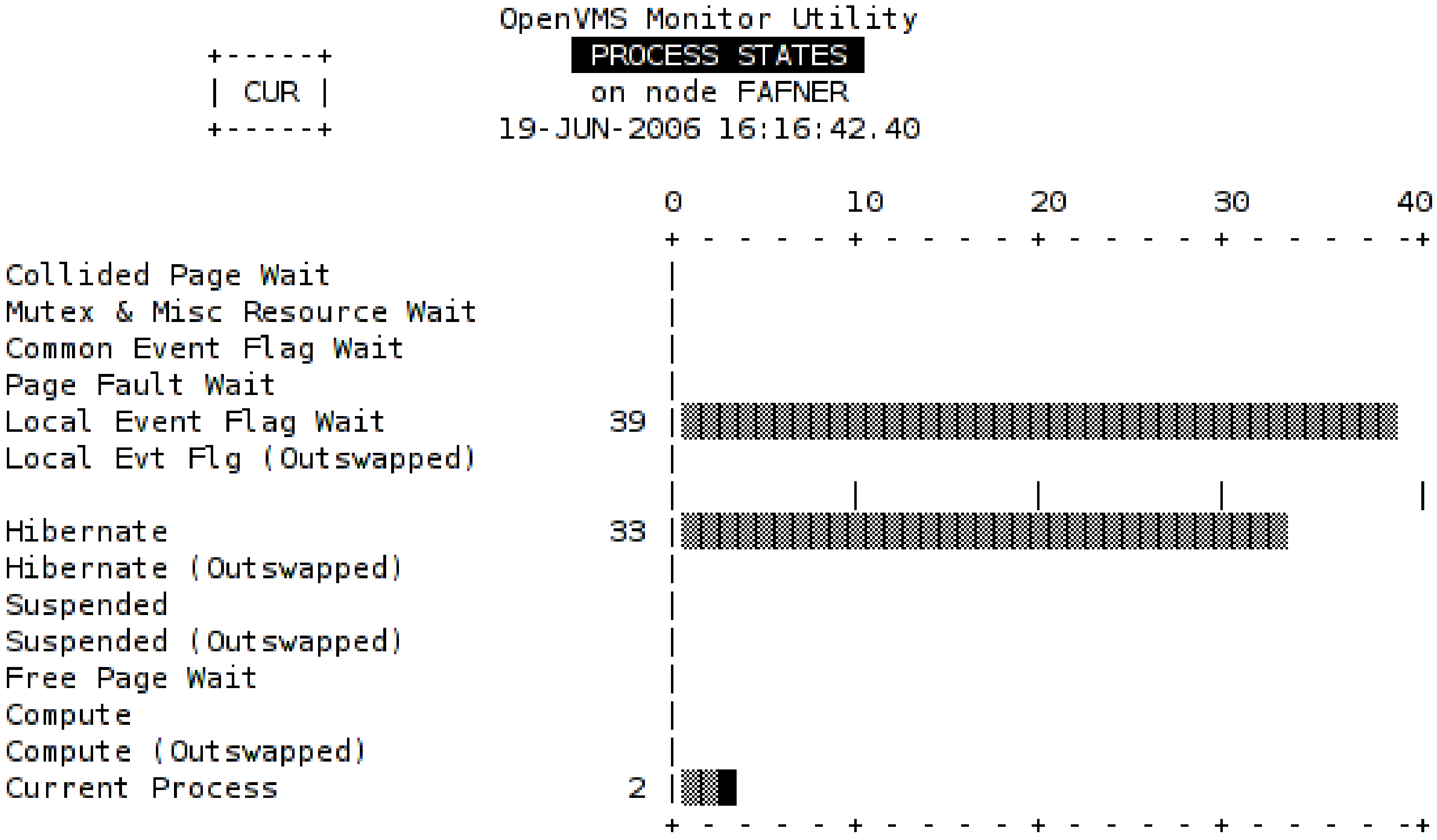
- □ $MONITOR SYSTEM



```
Node: FAFNER                 OpenVMS Monitor Utility     18-JUN-2006 13:47:59
Statistic: CURRENT              SYSTEM STATISTICS
                                                            Process States
              ┌ CPU Busy (14)                ┐       LEF:     38        LEFO:     0
              │▒                             │       HIB:     33        HIBO:     0
CPU       0   ├──────────────────────────────┤ 200   COM:      0        COMO:     0
              │▒                             │       PFW:      0        Other:    1
              └──────────────────────────────┘       MWAIT:    0
              Cur Top: BATCH_799 (8)                             Total: 72

              ┌ Page Fault Rate (0)          ┐       ┌ Free List Size (481460) ┐
              ││                             │       │▒▒▒▒▒▒▒▒▒▒▒▒▒▒           │  873K
MEMORY    0   ├──────────────────────────────┤ 100 0 ├─────────────────────────┤
              │                              │       │▒▒▒▒▒▒▒                  │  142K
              └──────────────────────────────┘       └ Modified List Size (41351┘
              Cur Top:  (0)

              ┌ Direct I/O Rate (1)          ┐       ┌ Buffered I/O Rate (481) ┐
              │                              │       │▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒│
I/O       0   ├──────────────────────────────┤ 60  0 ├─────────────────────────┤  150
              │                              │       │▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒│
              └──────────────────────────────┘       └─────────────────────────┘
              Cur Top: BATCH_799 (1)                 Cur Top: BATCH_799 (450)
```

# Concepts: System Monitoring (4 of 4)

□ $MONITOR STATE



```
                                 OpenVMS Monitor Utility
             +-----+                 PROCESS STATES
             | CUR |                on node FAFNER
             +-----+             19-JUN-2006 16:16:42.40

                                      0    10   20   30   40
                                      +----+----+----+----+
   Collided Page Wait                 |
   Mutex & Misc Resource Wait         |
   Common Event Flag Wait             |
   Page Fault Wait                    |
   Local Event Flag Wait           39 |▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
   Local Evt Flg (Outswapped)         |
                                      |    |    |    |    |
   Hibernate                       33 |▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
   Hibernate (Outswapped)             |
   Suspended                          |
   Suspended (Outswapped)             |
   Free Page Wait                     |
   Compute                            |
   Compute (Outswapped)               |
   Current Process                  2 |▒█
                                      +----+----+----+----+
```

# Practical examples

□ timer example

□ submitting a job

□ submitting a job that re-submits itself

# References

- Duffy, Michael D. Getting Started with OpenVMS, Digital Press, 2003
- Hewlett-Packard. HP OpenVMS Systems. Available at http://h71000.www7.hp.com/. Accessed June 19, 2006.
- Miller, David Donald. OpenVMS System Concepts, 2nd ed. Digital Press, Boston, 1997.
- Wikipedia contributors. Files-11. Wikipedia, The Free Encyclopedia. June 12 2006, 12:09. Available at http://en.wikipedia.org/wiki/Files-11. Accessed June 19, 2006.
- Wikipedia contributors. OpenVMS. Wikipedia, The Free Encyclopedia. June 15 2006, 19:22. Available at http://de.wikipedia.org/wiki/OpenVMS. Accessed June 19, 2006.