# OpenVMS for Linux Users

Christof Ullwer

# Agenda

01 **Logging in**

02 **Getting help**

03 **Elementary File Tasks**

04 **Symbols and Logicals**

05 **Editors**

# Logging In

- Enter the username at the Username: prompt
- Enter the password at the Password: prompt
- For a user a secondary password can be defined
- OpenVMS usernames are usually uppercase but can be entered lowercase
- Passwords are case sensitive
- If the login fails the user might be disabled (login flag DISUSER is set)
- A maximum number of login failures automatically disables the user
- Access can be restricted to certain times on primary and/or secondary days
- A user account has default and authorized privileges as well as process parameter quotas and a default directory
- A welcome banner is displayed with information about last interactive and non-interactive logins as well as the number of login failures
- After a successful login a $ prompt is displayed indicating the DCL CLI

# DCL Basics

- At the $ prompt you can enter any command that is defined
- The prompt can be customized with SET PROMPT
- Unrecognized input results in the warning

```
%DCL-W-IVVERB, unrecognized command verb - check validity and spelling
```

- Commands can be abbreviated
- Executables have to be activated with the RUN command
- Define a symbolic name for a character string or integer value:
  symbol-name =[=] expression
- Define a symbolic name for a character string value:
  symbol-name :=[=] string
- A symbolic name can define an alias:
  e.g. LIST := DIRECTORY
- Using a $ sign in the expression or string defines an alias for running an executable
  e.g. LISP:=$DISK$USER:[XOF.LISP]LISP.EXE
- To delete a symbolic name use DELETE/SYMBOL

# DCL Basics

- Commands can be abbreviated
- Executables have to be activated with the RUN command
- Define a symbolic name for a character string or integer value:
  symbol-name =[=] expression
- Define a symbolic name for a character string value:
  symbol-name :=[=] string
- A symbolic name can define an alias:
  e.g. LIST := DIRECTORY
- Using a $ sign in the expression or string defines an alias for running an executable
  e.g. LISP:=$DISK$USER:[XOF.LISP]LISP.EXE
- To delete a symbolic name use DELETE/SYMBOL

# Getting Help

- HELP
- Output is automatically paged
- You enter at the top level and all topics are displayed
- Selecting a topic shows the details and a possible list of subtopics
- Selecting a subtopic takes you deeper into the tree
- Pressing ENTER takes you one level up
- Most topics have a Parameter and an Examples subtopic

# Directory Structure

**NODE::DEVICE:[DIR1.SUBDIR1]FILENAME.EXTENSION;VERSION**

- E.g. IMPETU::DKA0:[SYS0.SYSMGR]SYSTARTUP_VMS.COM;9
- Special directory [000000] or [0,0]
- Logical names can be more expressive and simplify navigation
- ASSIGN creates a logical name and assigns an equivalence string, or a list of strings, to the specified logical name, DEFINE associates an equivalence name with a logical name
- There are different tables for logical names
- Can be listed with SHOW LOGICAL:

```
$ SHOW LOGICAL SYS$SYSTEM
   "SYS$SYSTEM" = "SYS$SYSROOT:[SYSEXE]" (LNM$SYSTEM_TABLE)

$ SHOW LOGICAL SYS$SYSROOT
   "SYS$SYSROOT" = "DKA0:[SYS0.]" (LNM$SYSTEM_TABLE)
         = "SYS$COMMON:"
1  "SYS$COMMON" = "DKA0:[SYS0.SYSCOMMON.]" (LNM$SYSTEM_TABLE)
```

# Elementary File Tasks

- DIRECTORY [/FULL] [/PROTECTION] [/OWNER] [/SIZE] [/DATE]
- SET DEFAULT
- COPY [/LOG] FILE%.PAS,FILE2.OBJ,FILE3.EXE [TMP]
  COPY [SMITH]MONKEY.DIR [JONES]
  COPY [SMITH.MONKEY]*.* [JONES.MONKEY]*.*
- RENAME  AVERAGE.OBJ  OLDAVERAGE
  RENAME WATER.TXT [.MEMOS]
  RENAME  [BORDERS.TESTFILES]SAVE.DAT  []TEST
- DELETE [/CONFIRM] [/TREE] [/LOG]
- PURGE [/KEEP] [/LOG]
- SET FILE /PROTECTION
- SHOW SECURITY

# Wildcards

Explained by Example:

- DELETE *.*;*
- COPY SQLQUERY%.SQL [.SUBDIR]
- RENAME *.LSP *.LISP
- DIRECTORY [...]
- SET DEFAULT [---]

# Editors

- EDIT/EDT
- EDIT/TPU
- EMACS
- LSE

# Demo!

# Thanks!

Q&A

# OpenVMS for Linux users

Veli Körkkö / Verticom

# Guidelines

Agenda

# Key system startup files

- SYS$MANAGER:SYCONFIG.COM
- SYS$MANAGER:SYLOGICALS.COM
- SYS$MANAGER:SYPAGSWPFILES.COM
- SYS$MANAGER:SYSECURITY.COM
- SYS$MANAGER:SYSTARTUP_VMS.COM

And one not SHUTDOWN file as well.

- SYS$MANAGER:SYSHUTDWN.COM

# SYCONFIG

# SYS$MANAGER:SYCONFIG.COM

```
$!      This is an empty site-specific device configuration procedure.
$!      If you have specific device configuration requirements at your
$!      site, you should place the required commands in this procedure.
$!
$!      Use caution when placing configuration commands in this file
$!      which are only to be executed on specific nodes. By default
$!      SYCONFIG.COM is in the cluster-common directory SYS$COMMON
$!      and is therefore executed by all nodes.
$!
$!
$!      Remove the comment from the line below if you wish to prevent
$!      VMS from configuring devices with a SYSMAN IO AUTOCONFIGURE
$!      command.  This is typically only required if you are debugging
$!      new device drivers.
$!
$!      STARTUP$AUTOCONFIGURE_ALL == 0
```

SYCONFIG

# SYS$MANAGER:SYCONFIG.COM

```
$ set noon
$!      STARTUP$AUTOCONFIGURE_ALL == 0
$ if .not. f$getdvi("VTA0:","EXISTS")
$ then
$  mcr sysman io connect vta0 /noadapter /driver=sys$ttdriver.exe
$ endif
$!
```

# SYLOGICALS

# SYS$MANAGER:SYLOGICALS.COM

```
..
$! DEFINE/SYSTEM/EXECUTIVE SYSUAF          SYS$SYSTEM:SYSUAF.DAT
$! DEFINE/SYSTEM/EXECUTIVE SYSUAFALT       SYS$SYSTEM:SYSUAFALT.DAT
$! DEFINE/SYSTEM/EXECUTIVE SYSALF          SYS$SYSTEM:SYSALF.DAT
$! DEFINE/SYSTEM/EXECUTIVE RIGHTSLIST      SYS$SYSTEM:RIGHTSLIST.DAT
$! DEFINE/SYSTEM/EXECUTIVE NETPROXY        SYS$SYSTEM:NETPROXY.DAT
$! DEFINE/SYSTEM/EXECUTIVE NET$PROXY       SYS$SYSTEM:NET$PROXY.DAT
...
```

This file is typically used for defining locations of SYSUAF, RIGHTSLIST etc. key system files. E.q. SYSUAF is basically your /etc/passwd and resides in SYS$SYSTEM as SYSUAF.DAT but you can have this and all others relocated.

Absolutely needed stuff for MULTI SYSTEM DISK clusters!

# SYLOGICALS

## SYS$MANAGER:SYLOGICALS.COM

The key reason for defining location of, say SYSUAF, RIGHTLIST etc. is e.g. to avoid following:

$ show logical sysuaf

%SHOW-S-NOTRAN, no translation for logical name SYSUAF

$ mcr authorize

%UAF-E-NAOFIL, unable to open system authorization file (SYSUAF.DAT)

-RMS-E-FNF, file not found

Do you want to create a new file?

$ set def sys$system:

$ mcr authorize

UAF>

# SYS$MANAGER:SYPAGSWPFILES.COM

This file would be needed with custom content in case you need to have custom setup of page and swap files.

SYS$SPECIFIC:[SYSEXE]SWAPFILE.SYS (not applicable for x86 systems)

and

SYS$SPECIFIC:[SYSEXE]PAGEFILE.SYS

will be automatically installed during startup.

# SYS$MANAGER:SYPAGSWPFILES.COM

```
$ show mem/files
                System Memory Resources on 24-FEB-2026 12:38:18.87

Paging File Usage (8KB pages):                  Index        Free        Size
  DISK$SYS_JUHANI:[SYS0.SYSEXE]PAGEFILE.SYS;2
                                                  254     1000000     1000000
  Total committed paging file usage:                                    23882
```

- This shows a typical ”out of box” setup and no need for custom SYPAGSWPFILES.COM
- The actual size of file though approximately 16 times of show here as size of these files in shown as ”VMS PAGES” instead of disk BLOCKS

$ dir/Siz=all sys$system:*file.sys/noheader/notrailer/wid=file=40

SYS$SYSROOT:[SYSEXE]PAGEFILE.SYS;2  16000128/16000128

# SYS$MANAGER:SYPAGSWPFILES.COM

```
$ show mem/files
             System Memory Resources on 24-FEB-2026 12:41:28.70

Swap File Usage (8KB pages):                    Index        Free        Size
  DISK$I64SYS:[SYS0.SYSEXE]SWAPFILE.SYS
                                                    1        6456        6456

Paging File Usage (8KB pages):                  Index        Free        Size
  DISK$I64SYS:[SYS0.SYSEXE]PAGEFILE.SYS
                                                  254     1572856     1572856
  Total committed paging file usage:                                      9913
$
```

- This shows a typical ”out of box” setup and no need for custom SYPAGSWPFILES.COM. Same comment about actual file size.
- But here (personal opinion) it’s ”bad setup” as SWAPFILE.SYS is way too small.
- Better to resize it properly or even get rid of it completely.

# SYS$MANAGER:SYSECURITY.COM

```
$ ty sys$manager:sysecurity.com
$!
$! This command procedure is run prior to starting up the security auditing
$! server process. Its purpose is to mount or define any disks which will be
$! used to hold security auditing log files (primarily the system security
$! audit journal file) or local security archive files.
$!
```

*Personally have never used this file.*

# SYS$MANAGER:SYSTARTUP_VMS.COM

This file is really the main file to be edited by SYSTEM MANAGER aka SYSADMIN. A first few lines of what I typically have are shown:

```
$ set noon
$ set logins/interactive=0                    ! to allow logins for accounts with at least OPER privilege during startup
$ node = f$getsyi("NODENAME")                 ! To setup symbol containing our system nodename
$!
$ write sys$output ""
$ write sys$output "The OpenVMS system is now executing the site-specific startup commands."
$ write sys$output ""
```

# SYS$MANAGER:SYSTARTUP_VMS.COM

$ Define/System/Exec T4$SYS sys$sysdevice:[T4$SYS]

$ define/system/exec t4$data sys$sysdevice:[t4$data]

$!  Above commands probably could be in that SYLOGICALS.COM

$ define/system sys$batch 'node'_startup$batch

$ initialize/queue/batch/job=5/base=3/prot=(W)/on='node'::/start 'node'_startup$batch

Then this would setup a batch queue for some things better started as batch jobs whilst system is still struggling to get up.

# SYS$MANAGER:SYSTARTUP_VMS.COM

$ @sys$startup:ld$startup.com

$ @sys$startup:lm$startup.com

$ define/system tcpip$smtp_log_level 3

$ @sys$startup:tcpip$startup.com

$ @sys$startup:ssh$startup.com

Just to show some further stuff there.

# SYS$MANAGER:SYSTARTUP_VMS.COM

```
$!      **** system startup final tasks
$ define/system sys$batch       'node'$batch
$ enable autostart/queues
```

Above to redefine SYS$BATCH logical so that possible BATCH jobs would go to ”normal queue”

```
$ startup$interactive_logins == 32                    !to have interactive user limit at then od system startup
$ mail/subject=""node' completed SYSTARTUP_VMS at ''f$time()'"/noself nla0: system
$ exit 1
```

And the mail command would there notify me my system has completed startup.

# SYS$MANAGER:SYSHUTDWN.COM

```
$!      This is an empty site-specific system shutdown procedure.
$!      If you have specific system shutdown requirements at your
$!      site, you should place the required commands in this procedure.
$!
```

- Not a STARTUP related file, quite the reverse, optionally executed by SYS$SYSTEM:SHUTDOWN.COM during shutdown.

SYSHUTDWN

# SYS$MANAGER:SYSHUTDWN.COM

```
$ set noon
$!
$ file := SYS$STARTUP:VMSSPI$SHUTDOWN.COM
$ if f$search(""file"") .nes. "" then @'file’
```

# SYS$MANAGER:SYSHUTDWN.COM

```
$ @sys$system:shutdown
        SHUTDOWN -- Perform an Orderly System Shutdown
                on node JUHANI
```

- How many minutes until final shutdown [0]:
- Reason for shutdown [Standalone]:
- Do you want to spin down the disk volumes [NO]?
- Do you want to invoke the site-specific shutdown procedure [YES]?

MODPARAMS

# SYS$SYSTEM:MODPARAMS.DAT

Not a startup/shutdown file but very important system parameter file.

Contains stuff to be used by AUTOGEN tool to calculate key system parameters.

# MODPARAMS

# SYS$SYSTEM:MODPARAMS.DAT

```
! SYS$SYSDEVICE:[SYS0.SYSEXE]MODPARAMS.DAT
! Created during installation of OpenVMS x86_64 V9.2-1 10-JAN-2024 20:49:55.36
!
VAXCLUSTER            = 0
SCSNODE               = "JUHANI"
SCSSYSTEMID           = 64522       ! 63.10
!
mscp_load             = 0
tmscp_load            = 0
niscs_load_pea0       = 0
!
system_check          = 0
!
```

# MODPARAMS

## SYS$SYSTEM:MODPARAMS.DAT

```
alloclass          = 1
shadowing          = 2
shadow_sys_disk    = 1
shadow_sys_unit    = 100
!
pagefile           = 0
swapfile           = 0
dumpfile           = 0
dumpstyle          = 11
```

Here e.g. the line <pagefile, swapfile, dumpfile> instructs AUTOGEN not to mess up what I feel best. Eventually one will end up having tons of stuff here.

BACKUP utility

# BACKUP UTILITY

OpenVMS has a native BACKUP utility ($ HELP BACKUP) that can be used for so-called PHYSICAL mode, IMAGE mode and FILE mode backups and restores (of files or even entire disks).

- BACKUP can use as either source or target also so-called SAVESETs, containers for files.

- BACKUP can do compression and encryption as well (and reverse too).

BACKUP utility

# BACKUP /physical

This is basically almost like using dd command on Linux with the exception that either the source or target could also be a saveset (aka the container file) instead of just mere disk.

$ mount/override=ident $1$LDA2:

$ mount/foreign $1$LDA3:

$ backup/physical $1$LDA2: $1$LDA3:

Basically same as

# dd if=/dev/sdm of=/dev/sdn

BACKUP utility

# BACKUP /image

$ mount/override=ident $1$LDA2:

$ mount/foreign $1$LDA3:

$ backup/image $1$LDA2: $1$LDA3:

And now target disk is with same contents except that free space, files would be defragmented.

BACKUP utility

# BACKUP /image for backup

$ mount/override=ident $1$LDA2:

$ mount/override=ident $1$LDA4:

$ backup/image $1$LDA2: $1$LDA4:[backups]LDA2.BCK/Save

- And the target could be a tape drive as well.
- Now we have on the target media a full ”image mode” backup of our disk.

BACKUP utility

# BACKUP /image  for restoring

Let’s say I need to restore my LDA2:

$ dismount $1$LDA2:
$ mount/foreign $1$LDA2:
$ mount/override=ident $1$LDA4:
$ backup/image $1$LDA4:[backups]LDA2.BCK/Save $1$LDA2:
$ dismount $1$LDA2:
$ mount/system $1$LDA2: test

And that LDA2.BCK could be from tape media as well.

# BACKUP /image for restores

Note that whilst IMAGE mode restore always needs a saveset made in IMAGE mode for source, you can also restore just files from that saveset made with BACKUP/IMAGE. E.g.

```
$ mount/foreign $1$MKA500:
$ backup $1$MKA500:LDA2.BCK/save/select=[korkko.tools]*.com –
        $1$LDA2:[korkko.toolsfromtape]
```

This saveset LDA2.BCK could also have been in file mode.

BACKUP utility

# BACKUP for file mode operations

Both source and target can be disks or one can be tape media (mounted foreign).

$ backup/log $1$LDA2:[KORKKO…]*.*;* $1$LDA3:[VELI…]

$ backup/log $1$LDA2:[KORKKO…]*.*;* $1$LDA3:[backups]KORKKO.BCK/sav

Reverse

$ backup/log $1$LDA3:[backups]KORKKO.BCK/sav $1$LDA2:[korkko…]

# Thanks!