

Seminar

Betriebssystemdienste & -administration



OpenVMS DCL

DIGITAL Command Language

Tobias Vogel

Torben Schreiter

Gliederung

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- Geschichte von OpenVMS
- DCL – *DIGITAL Command Language*
- OpenVMS – Features
 - Batch Jobs
 - Ressourcenmanagement
 - File System-Versionierung & -Security
- Zusammenfassung
- Literatur

Geschichte von OpenVMS

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- VMS – 1975 entwickelt von DEC (DIGITAL Equipment Corporation)
 - **V**irtual **M**emory **S**ystem
 - geschrieben für die VAX (32-Bit Architektur, CISC-Befehlssatz)
 - in Assembler
 - 1991 umbenannt in OpenVMS
 - 2 Portierungen auf die Alpha- & Intel Itanium-64 Bit Architekturen
- sehr stabile Laufzeitumgebung
- fortschrittliche Unterstützung von Clustering
- umfangreiches Ressourcenmanagement

DIGITAL Command Language

Einführung

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- Eingabesprache für die OpenVMS-Shell (CLI)
- Grundlegendes
 - alle Kommandos sind case insensitive
 - Eingabe-Prompt (normalerweise mit \$)
 - eine Anfrage besteht aus
 - einem Verb (z.B. **SHOW**)
 - einem optionalen Keyword (z.B. **QUEUE**)
 - optionalen Qualifiern (z.B. **/ALL_JOBS**)
 - optionalen Parametern (z.B. **SYS\$BATCH**)
 - DCL ist Vermittler zwischen der Eingabesyntax und dem eigentlichen Aufruf der Programme
 - es wird bei Aufruf eines Executables kein neuer Prozess erzeugt (**\$ SPAWN**) sondern das Image in den aktuellen Prozess geladen (DCL-Interpreter bleibt dabei resistent)

DIGITAL Command Language

Eingabe-Features

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- ausgelegt auf einfache Bedienbarkeit
 - Befehls-Abkürzungen benutzbar:
`$ SH SYS/F/PA=SA/SE="ULMANN"`
`$ SHOW SYSTEM/FULL/PAGE=SAVE/SEARCH="ULMANN"`
 - Parameter-Abfrage von obligatorischen Parametern
(Integration in die Shell; durch Command Tables)
 - Positional vs. Global Qualifiers
`$ PRINT/COPIES=2 A.TXT,B.TXT,C.TXT/COPIES=3`
 - Zeilenumbruch innerhalb eines Befehls mit "–"
 - eingängige, hierarchische Strukturierung der Online-Hilfe auf Basis von Help-Libraries
 - auch: kontextbasierte Aufbereitung (`$ HELP HINT`)

DIGITAL Command Language

Physische Geräte

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- Pfadstruktur:
NODE : : LOGICAL_NAME : [DIR . SUBDIR] FILE . EXT ; VERSION
- Dateisystem *Files-11 ODS-2*
 - maximale Länge des Dateinamens: 39.39
 - maximale Verzeichnistiefe: 8
 - Verzeichnisse sind Dateien (*.DIR*)
 - integriertes Versionierungssystem (wird später diskutiert)
 - keine zentrale Wurzel
- Magnetband-/CD-ROM-Laufwerke und andere Volumes werden montiert

DIGITAL Command Language

Symbole

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



○ frei definierbare Symbole

- jedes Terminal hat eigenen (globalen) Namensraum
- jedes command level hat zusätzlich einen eigenen lokalen Namensraum
- implizit getypte Variablen
 - unterschiedliche Zuweisungsoperatoren
 - := (lokal, impliziert String-Typ)
 - = (lokal, automatische Typzuweisung)
 - ::= (global, impliziert String-Typ)
 - == (global, automatische Typzuweisung)
- Beispiele: `cd`, `cls`, `eve`, `ls`
- arithmetische und String-Operationen mit Symbolen möglich

DIGITAL Command Language

Command Procedures

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



● Command Procedures (.COM)

- Zusammenfassung von DCL-Befehlen in einer Datei
- Verwendung von Steuerstrukturen möglich
 - Fallunterscheidungen
 - Schleifen
 - Subroutinen
- jede Zeile beginnt mit \$
- Kommentare mit !
- Ausführung durch \$ @**helloworld.com**
- Beispiel: login.com

DIGITAL Command Language

Command Definitions

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- Definition neuer DCL-Befehle für eigene Applikationen
- Eintragen der Befehle in *Command Tables*
 - system command table
`SYS$LIBRARY:DCLTABLES.EXE`
 - process command tables
- neue Befehle werden in Command Definition Files (.CLD) definiert
- Erweitern der *process command table* durch das Command Definition Utility (CDU)
 - `$ SET COMMAND`

DIGITAL Command Language

Help Libraries

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- Erstellung einer DCL-konformen Dokumentation zum Programm
(aufgerufen durch `$ HELP PRIM_P`)
- DCL-Hilfe ist baumartig strukturiert
- Anlegen einer .HLP-Datei mit den Text-Informationen
- Generierung einer Help-Library (.HLB) aus der .HLP-Datei
- Anschließend Einbinden in die systemweite Hilfe

OpenVMS – Features

Batch Jobs

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- Start von Command Procedures als Hintergrund-Batch Jobs möglich
- Ausführung mit `$ SUBMIT BATCH.COM`
- Batch-Verarbeitung basiert auf Queues mit festgelegter Basis-Priorität und Maximal-Länge
- Jobs haben eindeutige ID
- es wird ein LOG-File mit den Ergebnissen sowie statistischen Informationen erstellt

OpenVMS – Features

Ressourcenmanagement

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- umfangreiche Möglichkeiten der Ressourcenüberwachung
- zusätzliche spezifische Anpassungen möglich
 - Working Set für einen laufenden Prozess ändern
 - Priorität eines Prozesses ändern
 - Batchprozessen Maxima an CPU-Time oder Working Set-Größen zuweisen

OpenVMS – Features

File System-Versionierung & -Security

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- bei schreibendem Zugriff auf Dateien werden Kopien angelegt, anstatt die alte Datei zu überschreiben
FILE.EXT;VERSION+1
- Benutzung des Dateinamens ohne Version impliziert aktuellste Version
- Entfernen von älteren Versionen mit
\$ PURGE
- maximale Versionsanzahl pro Datei kann für Verzeichnisse angegeben werden
\$ SET DIRECTORY/VERSION_LIMIT=x

OpenVMS – Features

File System-Versionierung & -Security

Geschichte von OpenVMS
DIGITAL Command Language
OpenVMS – Features
Zusammenfassung
Literatur



- erteilbare Rechte:
Read+Write+Execute+Delete[+Control]
- Sicherheitseinstellungen können definiert werden über:
 - File Protection Bits
 - für SYSTEM, OWNER, GROUP und WORLD
 - vergleichbar mit UNIX rwx-Bits
 - **\$ SET SECURITY/PROTECTION=...**
 - Access Control Lists (ACLs)
 - haben Vorrang vor den Protection Bits
 - ermöglichen wesentlich feinere Zugriffssteuerung
 - ACEs als DEFAULT für Verzeichnisse definierbar
↳ werden für neue Dateien & Verzeichnisse übernommen
 - **\$ SET SECURITY/ACL=...**

Zusammenfassung



- DCL als Shell-Sprache sehr ausgereift
 - sinnvolle Features
 - einfach erweiterbar
 - integriertes Hilfesystem
- bot zur damaligen Zeit und auch heute herausragende Performanz (in Verbindung mit der Hardware)
- Fazit aus Praxisbeispielen (HP Success Stories):
 - VMS gelobt für Robustheit
 - Fähigkeit zur effektiven und flexiblen Clusterbildung
 - flexibles Ressourcenmanagement
- frühe Implementation vieler heute selbstverständlicher Konzepte



- DUFFY, Michael D.: Getting Started with OpenVMS: A Guide for New Users. Digital Press, 2003
- MILLER, David Donald: OpenVMS – Operating System Concepts, Second Edition. Digital Press, 1997
- RIECK, Neil: OpenVMS vs. UNIX (antagonists or sisters?), 2005
http://www3.sympatico.ca/n.riECK/docs/vms_vs_unix.html
- HP Corporation: OpenVMS Systems Documentation
<http://h71000.www7.hp.com/doc/>
- DCL Online Help



Fragen und Antworten