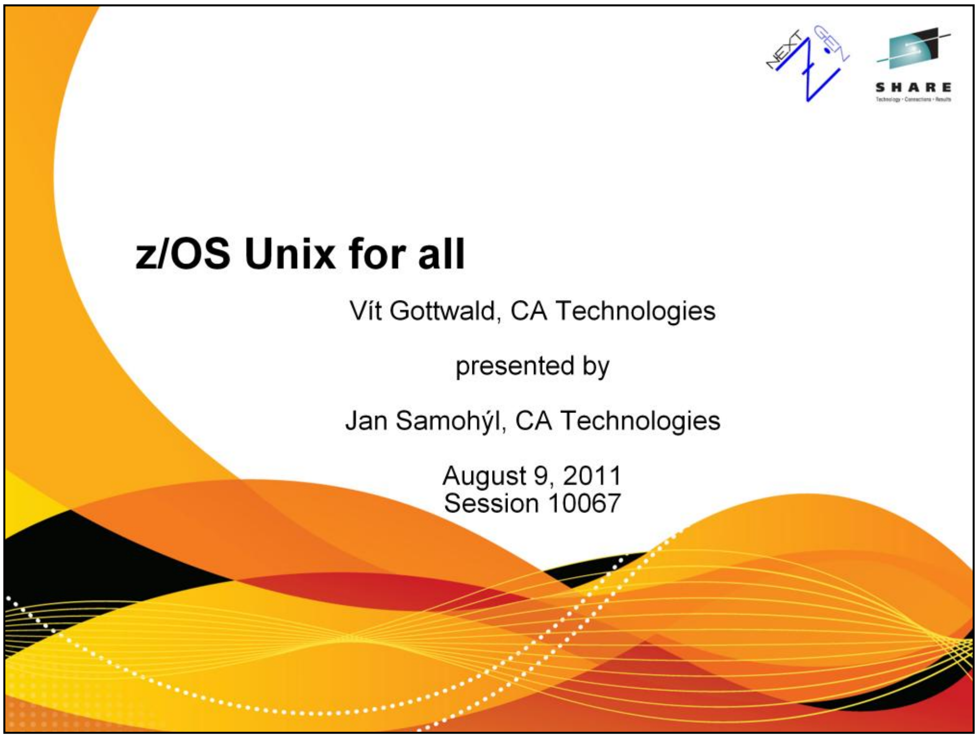


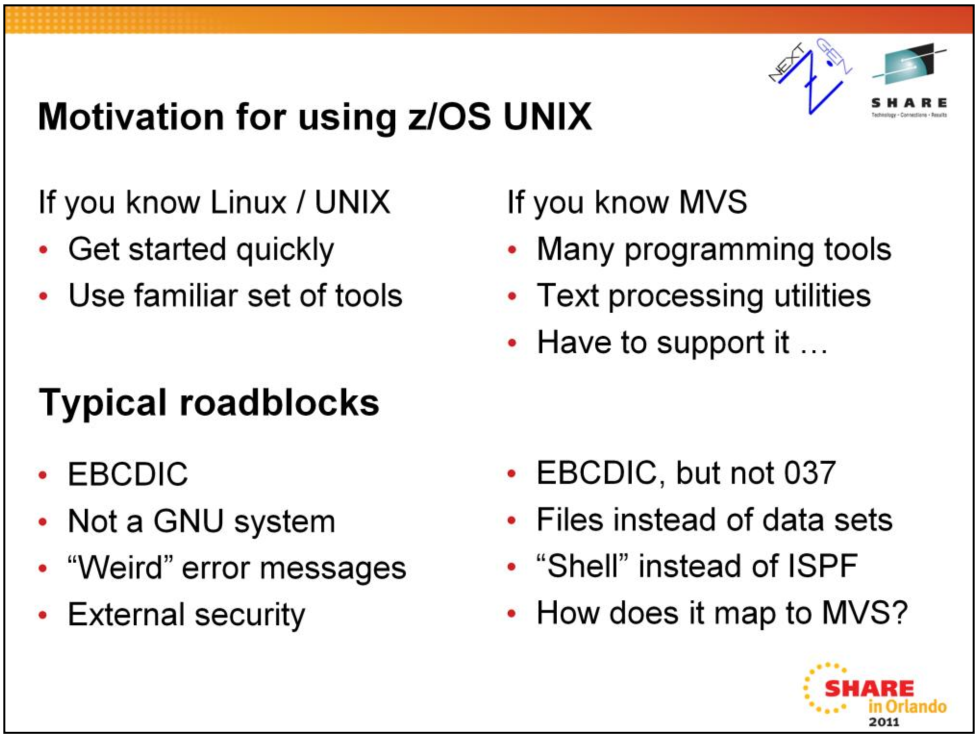
z/OS Unix for all

Vít Gottwald, CA Technologies

presented by

Jan Samohýl, CA Technologies

August 9, 2011
Session 10067



Motivation for using z/OS UNIX

If you know Linux / UNIX

- Get started quickly
- Use familiar set of tools

If you know MVS

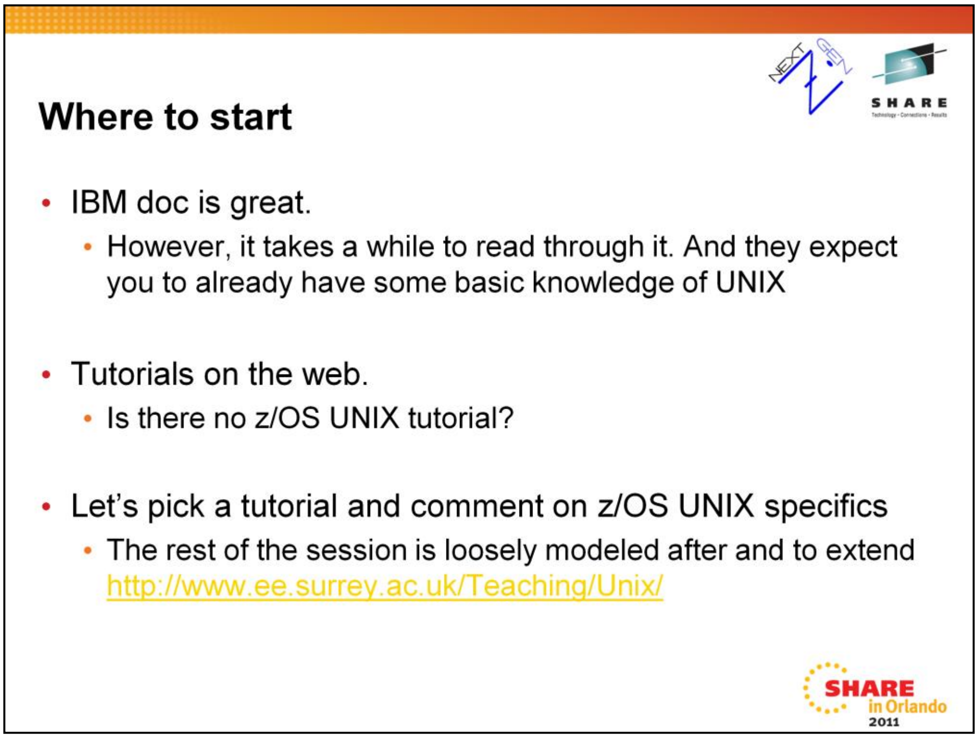
- Many programming tools
- Text processing utilities
- Have to support it …

Typical roadblocks

- EBCDIC
- Not a GNU system
- “Weird” error messages
- External security

- EBCDIC, but not 037
- Files instead of data sets
- “Shell” instead of ISPF
- How does it map to MVS?

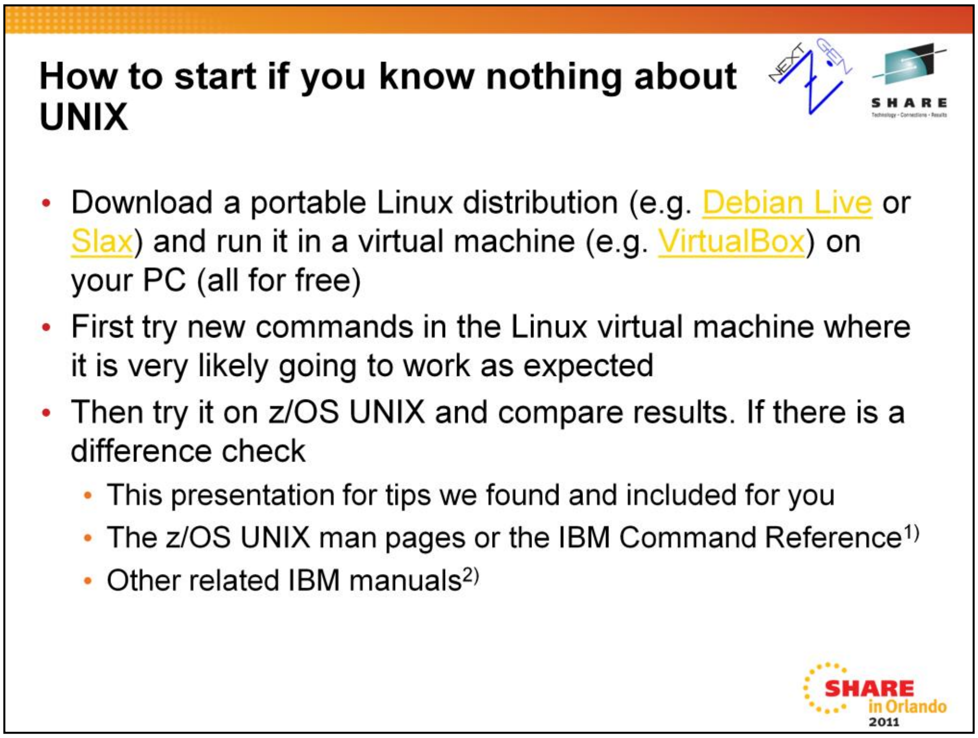
SHARE in Orlando 2011



Where to start

- IBM doc is great.
  - However, it takes a while to read through it. And they expect you to already have some basic knowledge of UNIX
- Tutorials on the web.
  - Is there no z/OS UNIX tutorial?
- Let's pick a tutorial and comment on z/OS UNIX specifics
  - The rest of the session is loosely modeled after and to extend http://www.ee.surrey.ac.uk/Teaching/Unix/

No z/OS UNIX tutorial? Well, we haven't seen one which is why we prepared this session.

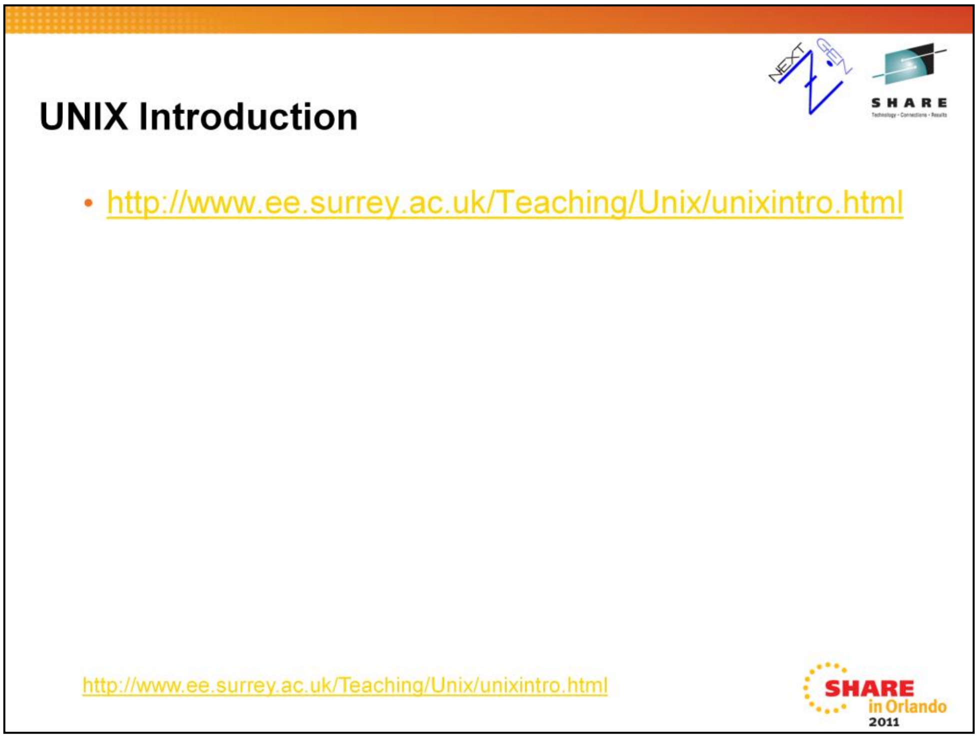


How to start if you know nothing about UNIX

- Download a portable Linux distribution (e.g. Debian Live or Slax) and run it in a virtual machine (e.g. VirtualBox) on your PC (all for free)
- First try new commands in the Linux virtual machine where it is very likely going to work as expected
- Then try it on z/OS UNIX and compare results. If there is a difference check
  - This presentation for tips we found and included for you
  - The z/OS UNIX man pages or the IBM Command Reference[1)]
  - Other related IBM manuals[2)]

1) The z/OS UNIX Command Reference SA22-7802 is an equivalent of UNIX manual pages. It is basically a concatenation of all the man pages into a single document plus the description of related TSO commands.

1) See "IBM z/OS UNIX manuals" slide at the end of this presentation

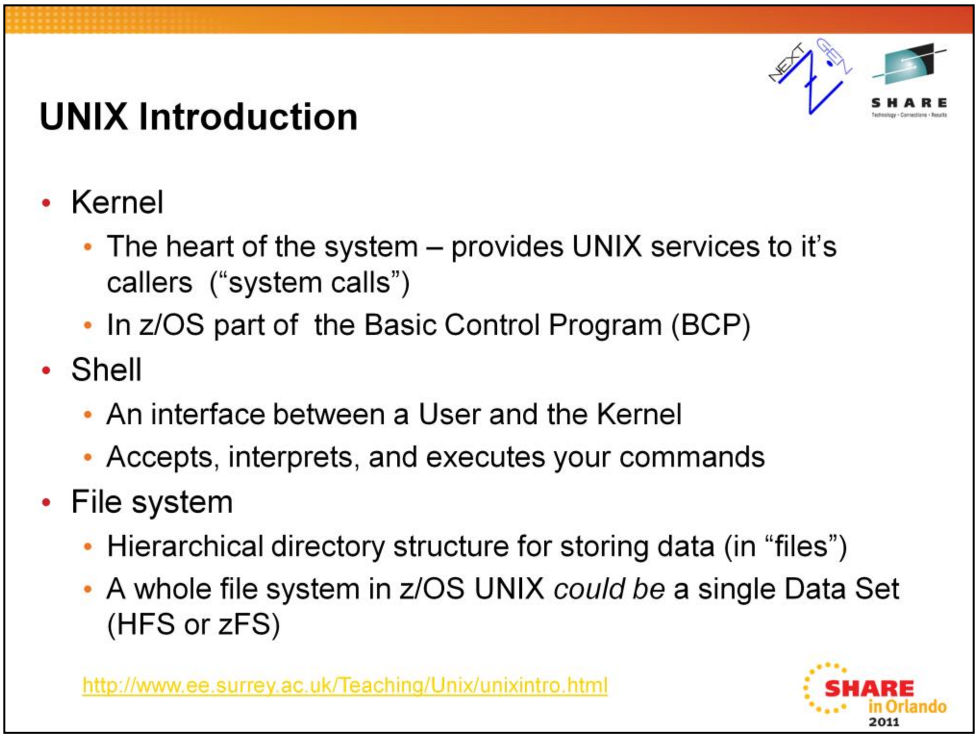


UNIX Introduction

- http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html

http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html

SHARE in Orlando 2011



UNIX Introduction

- Kernel
  - The heart of the system – provides UNIX services to it's callers ("system calls")
  - In z/OS part of the Basic Control Program (BCP)
- Shell
  - An interface between a User and the Kernel
  - Accepts, interprets, and executes your commands
- File system
  - Hierarchical directory structure for storing data (in "files")
  - A whole file system in z/OS UNIX *could be* a single Data Set (HFS or zFS)

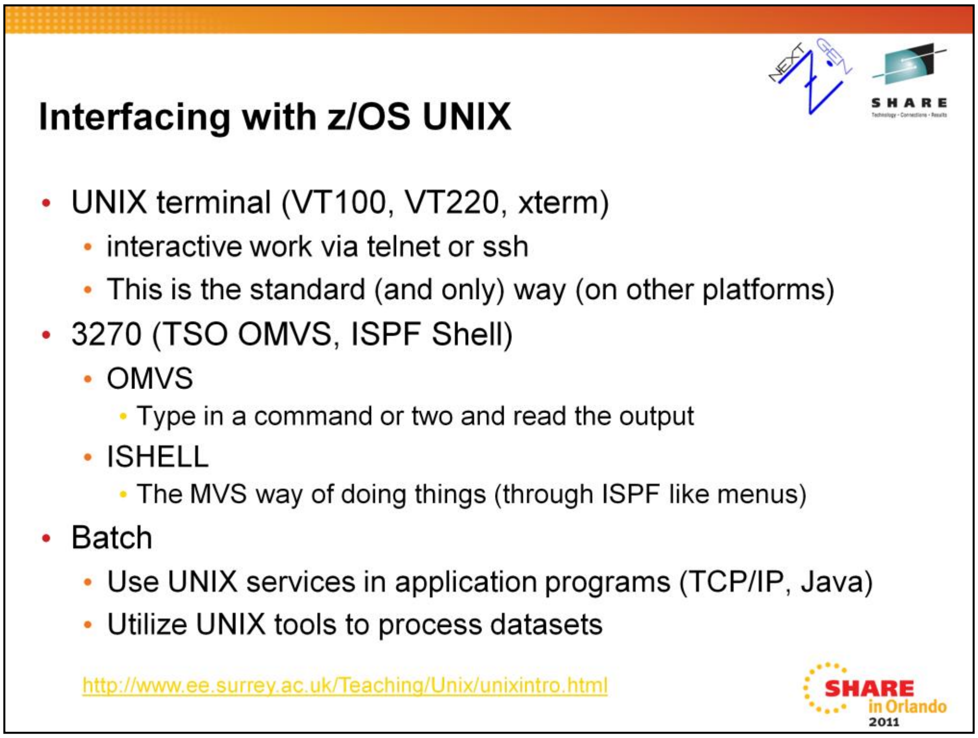
http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html

SHARE in Orlando 2011

could be – but typically is not, multiple datasets are used to have a single dataset for each logical part of the file system, see "Planing" manual or attend Session 9875: "z/OS Basics: The z/OS UNIX Shared File System Environment and How It Works" by Jim Showalter

Another essential part of the UNIX environment is formed by application programs. Especially the standard set of programs available on every UNIX, often called "commands" or "utilities".

From now on if appropriate there will be a link to the relevant part of the web tutorial at the bottom of each page.



Interfacing with z/OS UNIX

- UNIX terminal (VT100, VT220, xterm)
  - interactive work via telnet or ssh
  - This is the standard (and only) way (on other platforms)
- 3270 (TSO OMVS, ISPF Shell)
  - OMVS
    - Type in a command or two and read the output
  - ISHELL
    - The MVS way of doing things (through ISPF like menus)
- Batch
  - Use UNIX services in application programs (TCP/IP, Java)
  - Utilize UNIX tools to process datasets

http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html

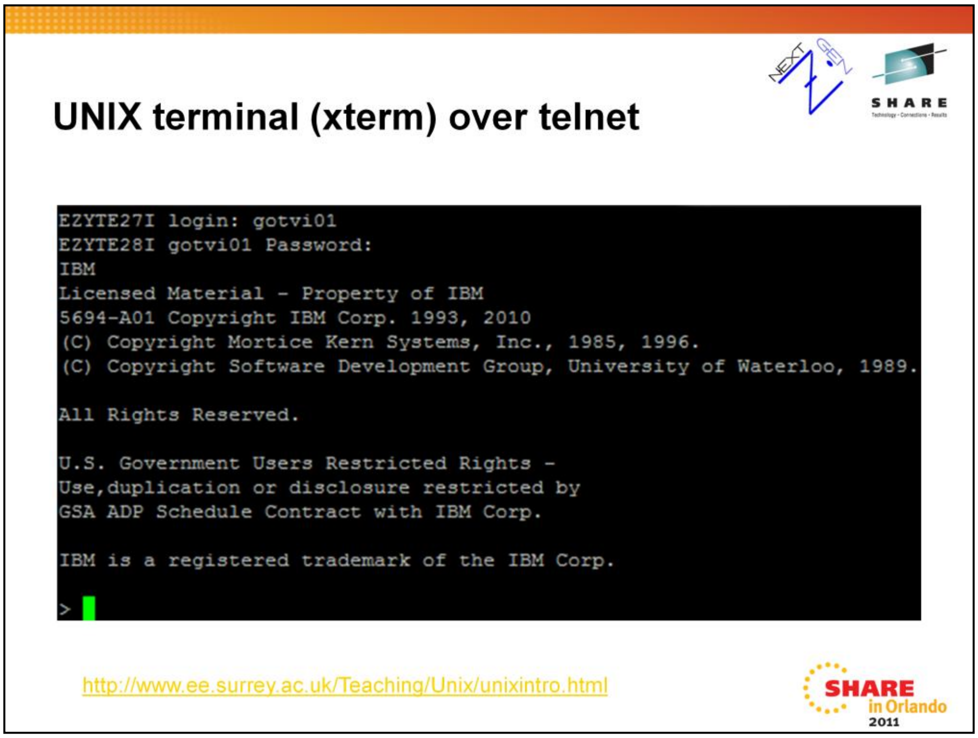
UNIX terminal modes:

a) Line at a time = line mode = canonical mode

b) Process every keystroke immediately = raw mode = non-canonical mode

3270 = screen at a time

In 3270 or vt100/xterm in line mode you loose many of the shell productivity features. Those could be a real difference between a pleasant and a mundane session and whether you develop a passion or a hatred for the system.



UNIX terminal (xterm) over telnet

```
EZYTE27I login: gotvi01
EZYTE28I gotvi01 Password:
IBM
Licensed Material - Property of IBM
5694-A01 Copyright IBM Corp. 1993, 2010
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government Users Restricted Rights -
Use,duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

>
```

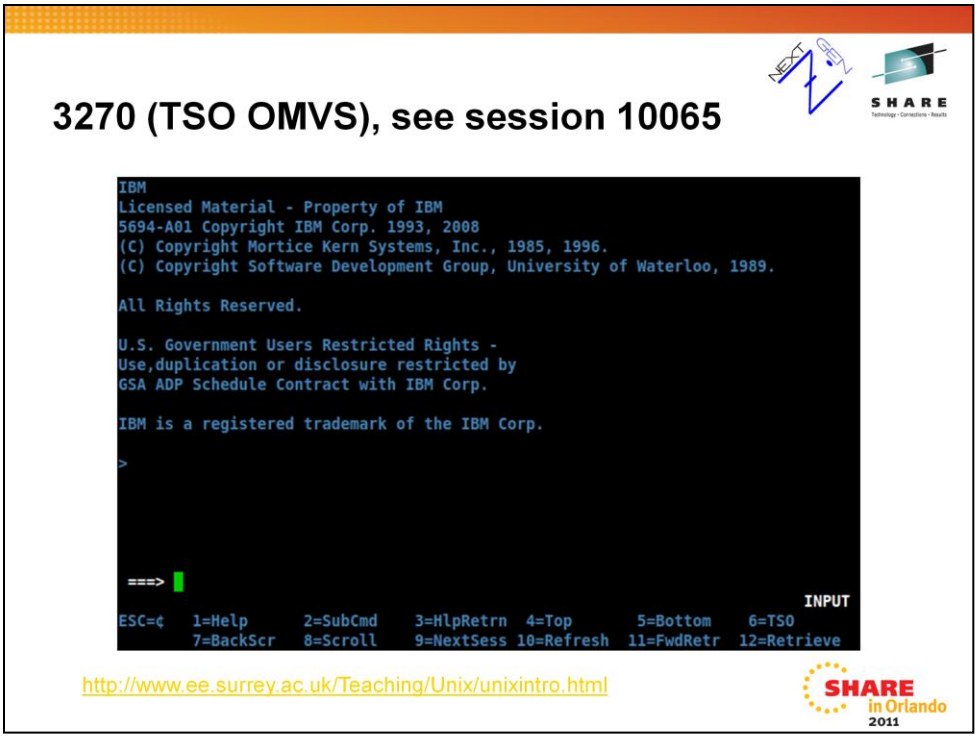
http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html

The de facto standard application for vt100/xterm emulation over telnet or ssh on the MS Windows platform is Putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/).

The UNIX terminal operates in one of two modes

1) Line / canonical – which sends line at a time to and from the UNIX box, each character received is immediately processed
2) Character / non-canonical / raw – every key pressed is sent to the UNIX box for processing and echoed back on the screen (unless echo is turned off)

Line mode is typically convenient if you are working on a system with response over 150 ms so that you don't have to wait for each individual keystroke to be processed before you can type the next one. You loose a lot of interactive features of the shell and other applications though (e.g. completions).



3270 (TSO OMVS), see session 10065

```
IBM
Licensed Material - Property of IBM
5694-A01 Copyright IBM Corp. 1993, 2008
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government Users Restricted Rights -
Use,duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

>




 ===>
                                                                    INPUT
ESC=¢    1=Help       2=SubCmd     3=HlpRetrn  4=Top       5=Bottom    6=TSO
         7=BackScr    8=Scroll     9=NextSess 10=Refresh  11=FwdRetr  12=Retrieve
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html

To start the session

enter “OMVS” from TSO command line or “TSO OMVS” from ISPF command line.

To quit the session:

a) type "exit" and hit ENTER

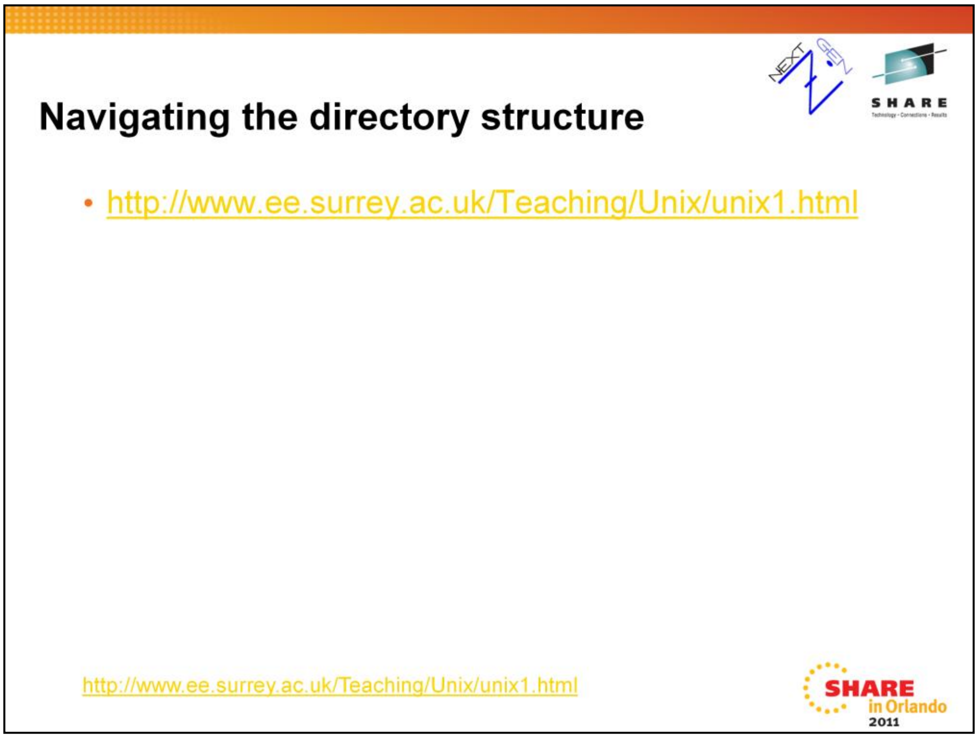
b) hit PF2 (SubCmd), then type in "quit" and hit ENTER

Are you curious what the ESC=¢ stands for?

To simulate Unix terminal's control sequences (like Ctrl-D) copy the character specified in ESC=¢ on the screen followed by the character (for Ctrl-D it is the 'D' character). In this case you want to send a Ctrl-D sequence, you will type into the command line "¢d" and then hit ENTER to send it for processing. This particular sequence will log you out of the session.

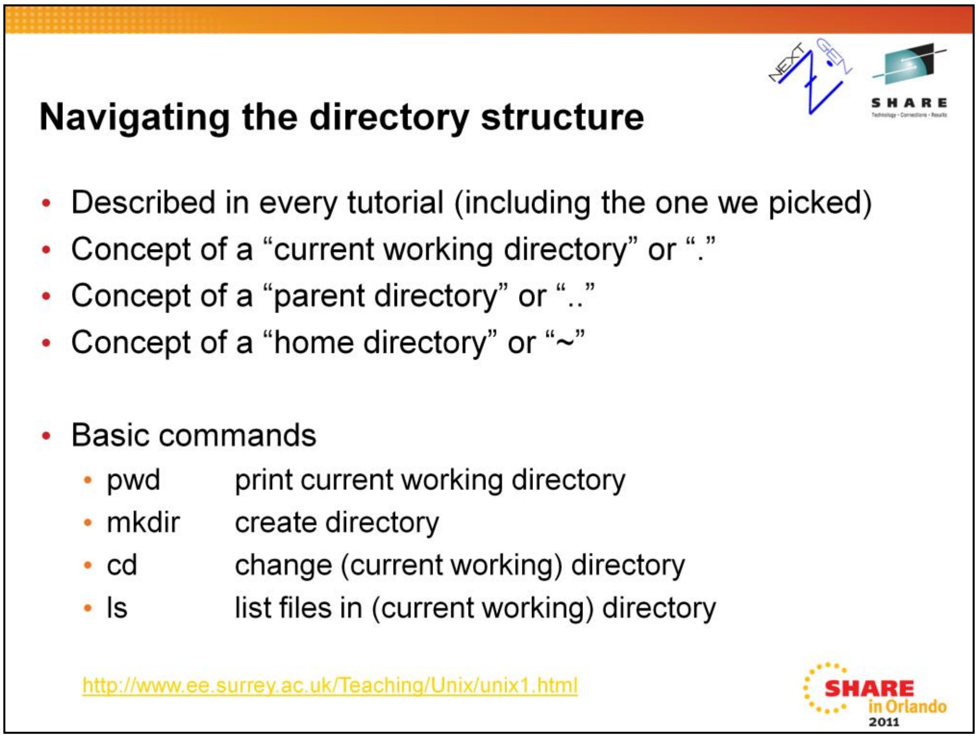
More about TSO OMVS in session **10065 - TSO OMVS and You: What to Make of UNIX System Services**

on Thursday, August 11, 2011: 11:00 AM-12:00 PM



Navigating the directory structure

- http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html



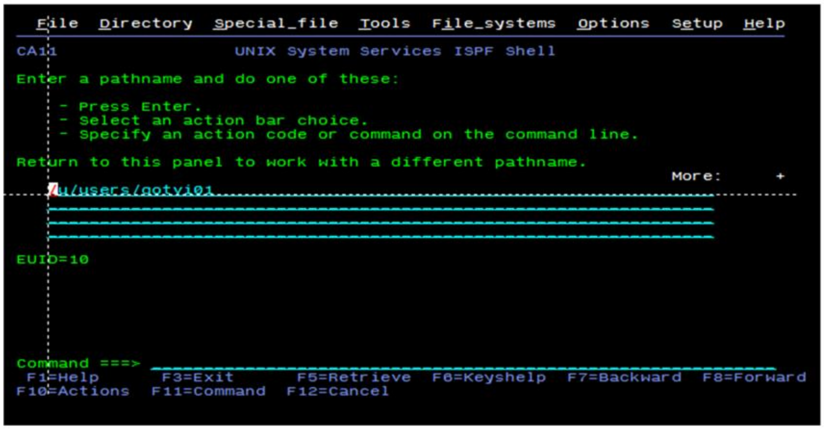
Navigating the directory structure

- Described in every tutorial (including the one we picked)
- Concept of a “current working directory” or “.”
- Concept of a “parent directory” or “..”
- Concept of a “home directory” or “~”

- Basic commands
  - pwd print current working directory
  - mkdir create directory
  - cd change (current working) directory
  - ls list files in (current working) directory

http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html

Navigating the directory structure, ISHELL – if you prefer it the MVS way

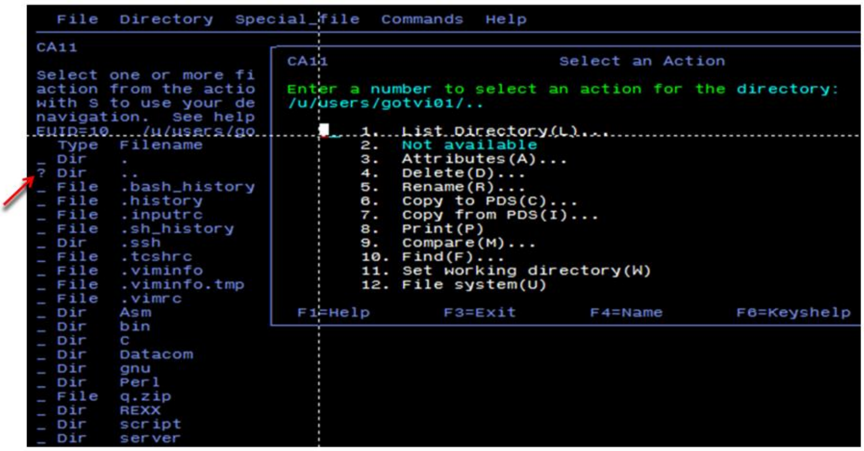


```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
CA11                    UNIX System Services ISPF Shell
Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                              More:     +
    /u/users/gotyi01

EUID=10

Command ===>
 F1=Help      F3=Exit      F5=Retrieve  F6=Keyshelp  F7=Backward  F8=Forward
F10=Actions  F11=Command  F12=Cancel
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html

Type ISHELL in your TSO session or TSO ISHELL from ISPF to invoke the ISPF Shell

Navigating the directory structure, ISHELL – if you prefer it the MVS way



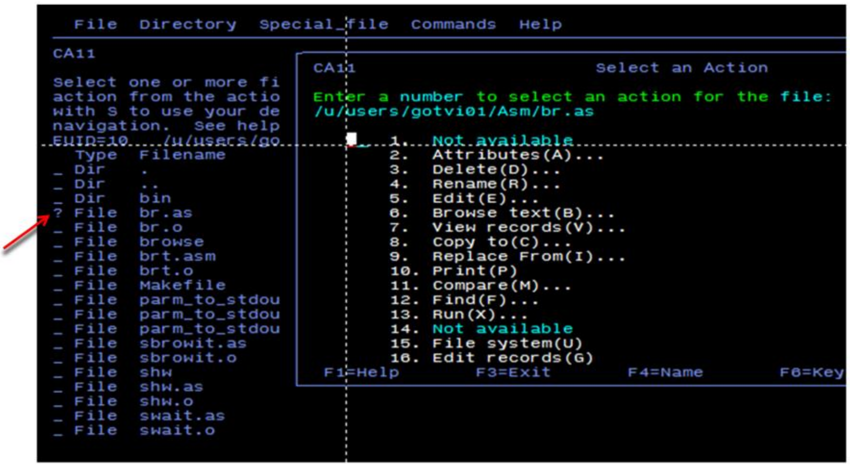
```
  File  Directory  Special_file  Commands  Help
 ─────────────────────────────────────────────────────────────────────────
 CA11
                          CA11                          Select an Action
 Select one or more fi
 action from the actio    Enter a number to select an action for the directory:
 with S to use your de    /u/users/gotvi01/..
 navigation.  See help
 EUID=10    /u/users/go        _  1.  List Directory(L)...
   Type  Filename                 2.  Not available
 _ Dir   .                        3.  Attributes(A)...
 ? Dir   ..                       4.  Delete(D)...
 _ File  .bash_history            5.  Rename(R)...
 _ File  .history                 6.  Copy to PDS(C)...
 _ File  .inputrc                 7.  Copy from PDS(I)...
 _ File  .sh_history              8.  Print(P)
 _ Dir   .ssh                     9.  Compare(M)...
 _ File  .tcshrc                  10. Find(F)...
 _ File  .viminfo                 11. Set working directory(W)
 _ File  .viminfo.tmp             12. File system(U)
 _ File  .vimrc
 _ Dir   Asm                F1=Help        F3=Exit        F4=Name        F6=Keyshelp
 _ Dir   bin
 _ Dir   c
 _ Dir   Datacom
 _ Dir   gnu
 _ Dir   Perl
 _ File  q.zip
 _ Dir   REXX
 _ Dir   script
 _ Dir   server
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html

Type a command next to a directory or a file to perform an action. A question mark will give you a menu of available actions.

This slides shows available actions for a directory.

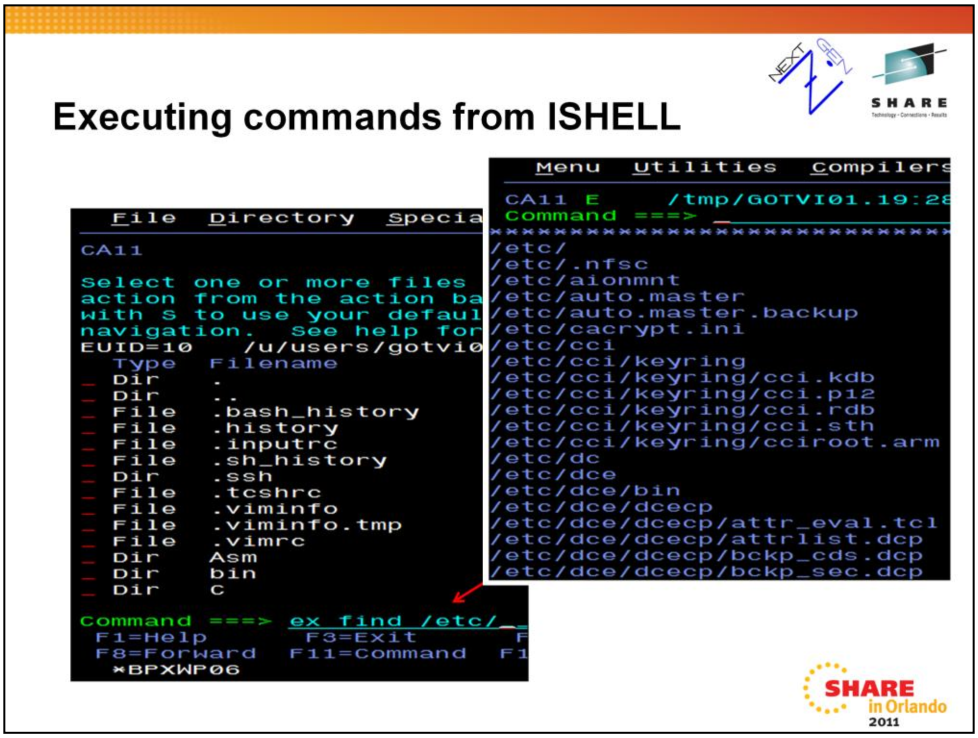
Navigating the directory structure, ISHELL – if you prefer it the MVS way



```
  File  Directory  Special_file  Commands  Help

CA11
                     CA11                    Select an Action
Select one or more fi
action from the actio Enter a number to select an action for the file:
with S to use your de /u/users/gotvi01/Asm/br.as
navigation.  See help
EUID=10    /u/users/go      _    1.  Not available
  Type  Filename                 2.  Attributes(A)...
_ Dir   .                        3.  Delete(D)...
_ Dir   ..                       4.  Rename(R)...
_ Dir   bin                      5.  Edit(E)...
? File  br.as                    6.  Browse text(B)...
_ File  br.o                     7.  View records(V)...
_ File  browse                   8.  Copy to(C)...
_ File  brt.asm                  9.  Replace From(I)...
_ File  brt.o                    10. Print(P)
_ File  Makefile                 11. Compare(M)...
_ File  parm_to_stdou            12. Find(F)...
_ File  parm_to_stdou            13. Run(X)...
_ File  parm_to_stdou            14. Not available
_ File  sbrowit.as               15. File system(U)
_ File  sbrowit.o                16. Edit records(G)
_ File  shw             F1=Help       F3=Exit       F4=Name       F6=Key
_ File  shw.as
_ File  shw.o
_ File  swait.as
_ File  swait.o
```

SHARE in Orlando 2011

This slides shows available actions for a file.



Executing commands from ISHELL

```
  File   Directory   Specia
CA11

Select one or more files
action from the action ba
with S to use your defaul
navigation.  See help for
EUID=10    /u/users/gotvi0
   Type   Filename
_  Dir    .
_  Dir    ..
_  File   .bash_history
_  File   .history
_  File   .inputrc
_  File   .sh_history
_  Dir    .ssh
_  File   .tcshrc
_  File   .viminfo
_  File   .viminfo.tmp
_  File   .vimrc
_  Dir    Asm
_  Dir    bin
_  Dir    C

Command ===> ex find /etc/_
 F1=Help       F3=Exit       F
 F8=Forward   F11=Command   F1
  *BPXWP06
```

```
   Menu   Utilities   Compilers
 CA11 E     /tmp/GOTVI01.19:28
 Command ===> _
*****************************
/etc/
/etc/.nfsc
/etc/aionmnt
/etc/auto.master
/etc/auto.master.backup
/etc/cacrypt.ini
/etc/cci
/etc/cci/keyring
/etc/cci/keyring/cci.kdb
/etc/cci/keyring/cci.p12
/etc/cci/keyring/cci.rdb
/etc/cci/keyring/cci.sth
/etc/cci/keyring/cciroot.arm
/etc/dc
/etc/dce
/etc/dce/bin
/etc/dce/dcecp
/etc/dce/dcecp/attr_eval.tcl
/etc/dce/dcecp/attrlist.dcp
/etc/dce/dcecp/bckp_cds.dcp
/etc/dce/dcecp/bckp_sec.dcp
```

SHARE in Orlando 2011

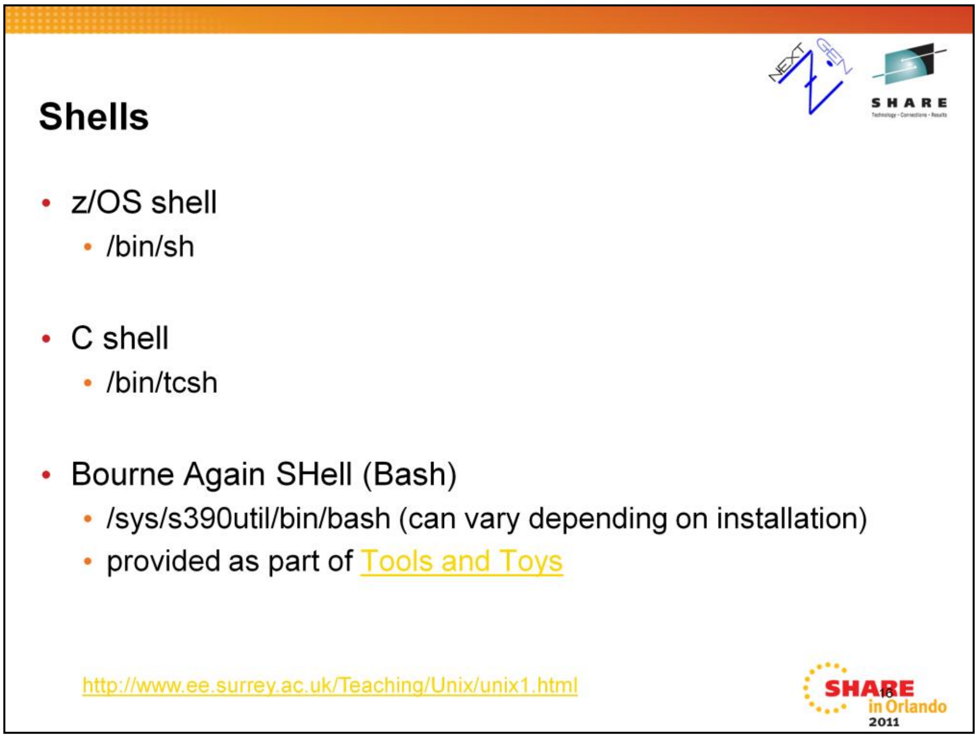
If you want to execute unix commands from ISHELL, type "sh command" or "ex command".

In the first case you will run your command within your shell {sh | tcsh | bash}. That means that /etc/profile and a profile in your home directory relevant to the particular shell will be processed before running the command.

In the latter case, the command will be run directly without any interaction with shell.

In both cases you will get the output in ISPF borwse like panel. You can scroll, search, etc.

find /etc/ will list all files in your /etc directory.



Shells

- z/OS shell
  - /bin/sh
- C shell
  - /bin/tcsh
- Bourne Again SHell (Bash)
  - /sys/s390util/bin/bash (can vary depending on installation)
  - provided as part of Tools and Toys

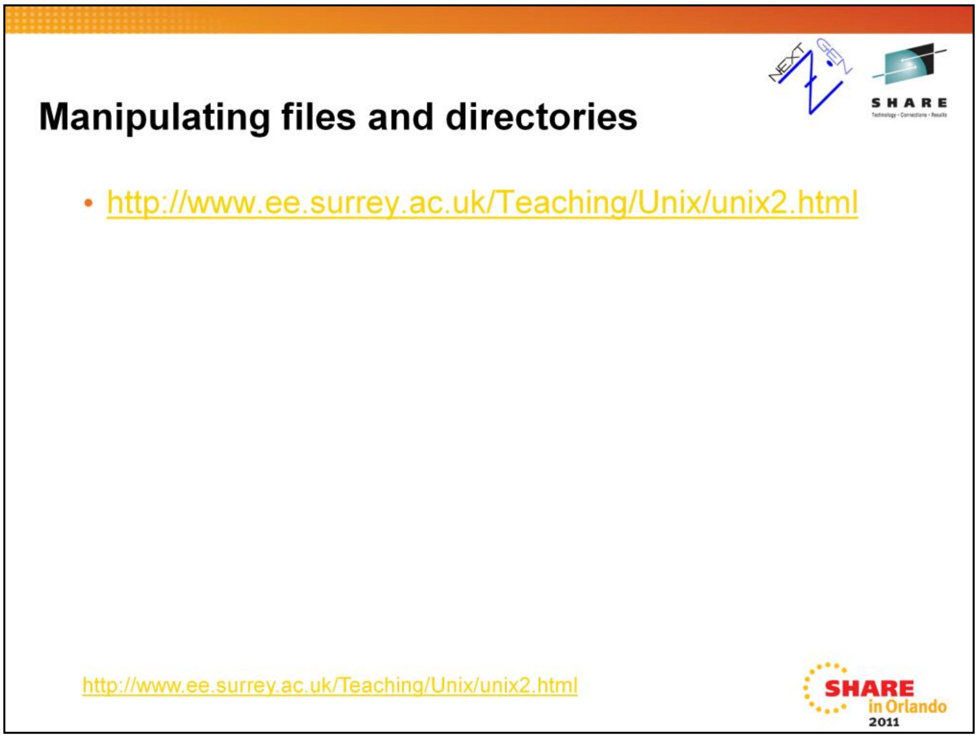
http://www.ee.surrey.ac.uk/Teaching/Unix/unix1.html

Shells can be used both for interactive work and batch processing.

z/OS shell comes from the Bourne Shell family. Unfortunately it is not user friendly over a telnet/ssh session.

On the other hand tcsh has been designed for interactive work and fits well to an interactive telnet/ssh sesssion.

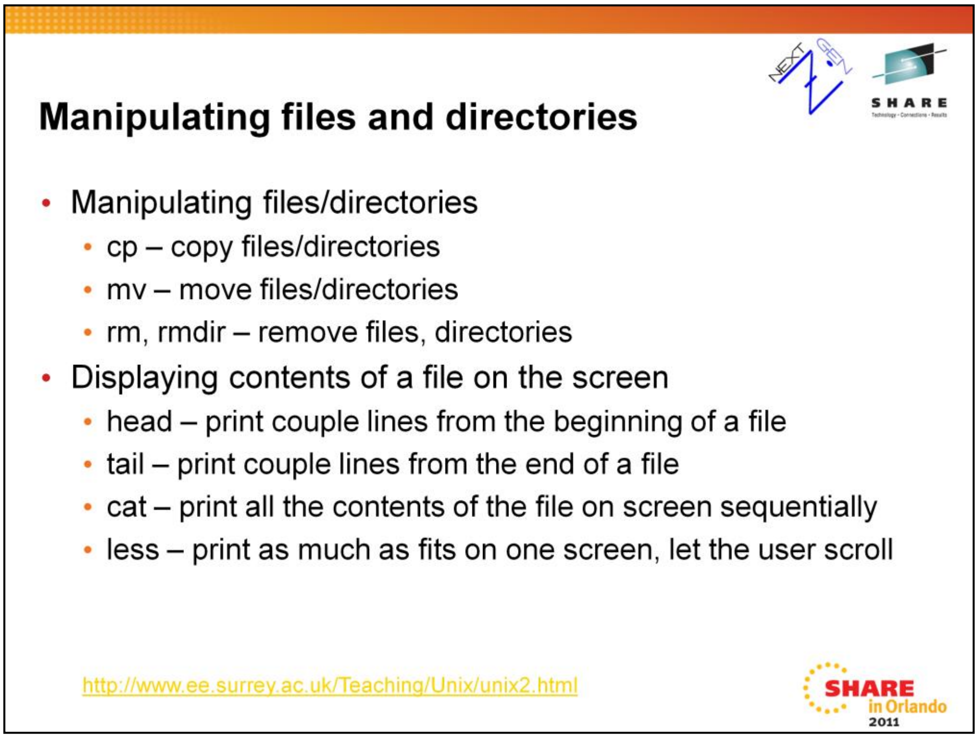
BASH (Bourne Again Shell) is a port of GNU BASH, a modern shell used in most Linux distributions. The z/OS UNIX port is not complete and not supported by IBM, but generally works as expected.



Manipulating files and directories

- http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

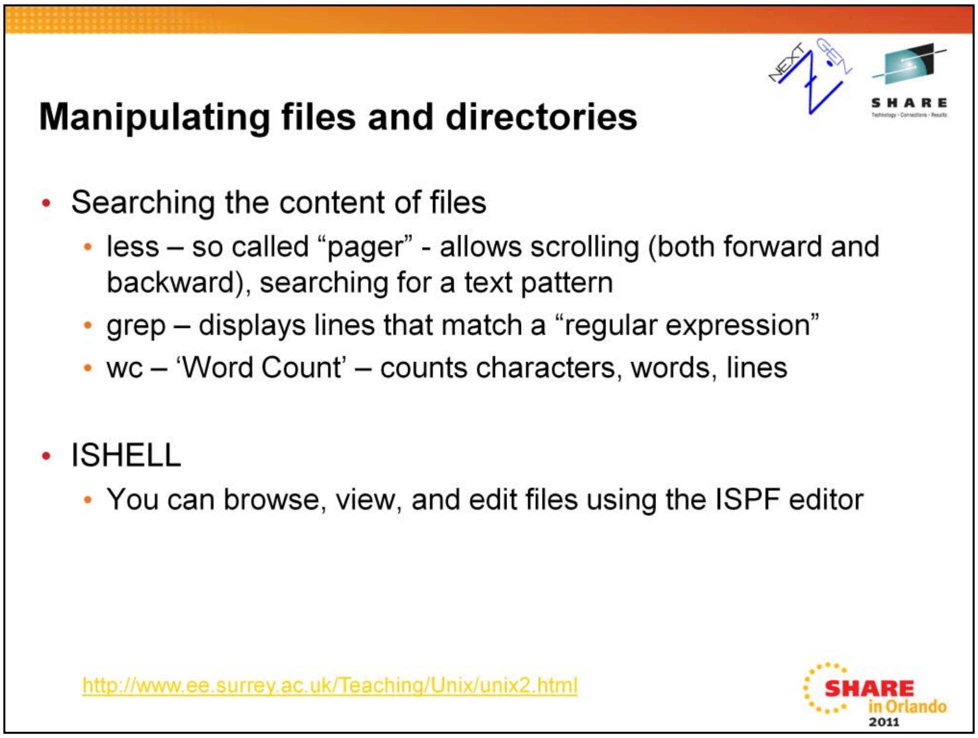
http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html



Manipulating files and directories

- Manipulating files/directories
  - cp – copy files/directories
  - mv – move files/directories
  - rm, rmdir – remove files, directories
- Displaying contents of a file on the screen
  - head – print couple lines from the beginning of a file
  - tail – print couple lines from the end of a file
  - cat – print all the contents of the file on screen sequentially
  - less – print as much as fits on one screen, let the user scroll

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

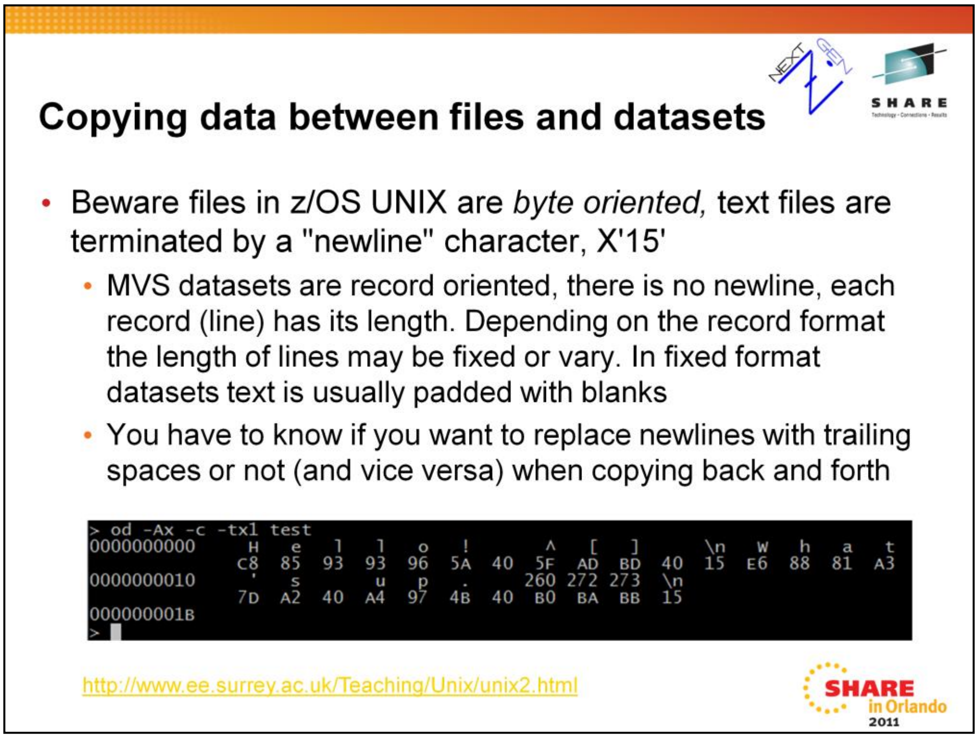


Manipulating files and directories

- Searching the content of files
  - less – so called “pager” - allows scrolling (both forward and backward), searching for a text pattern
  - grep – displays lines that match a “regular expression”
  - wc – ‘Word Count’ – counts characters, words, lines
- ISHELL
  - You can browse, view, and edit files using the ISPF editor

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

SHARE in Orlando 2011



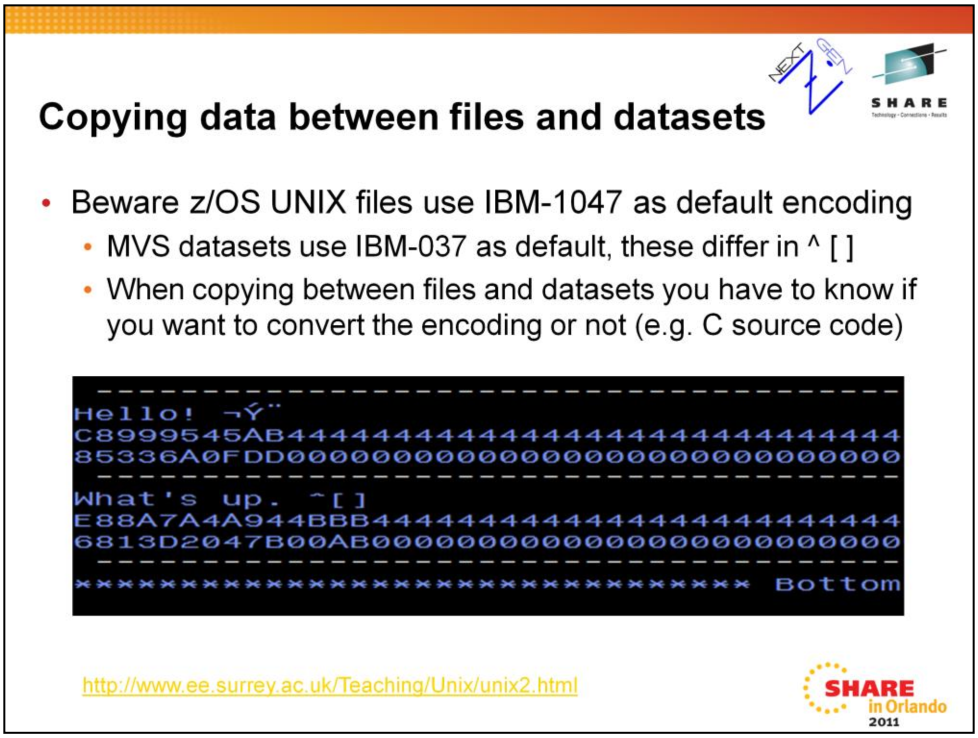
Copying data between files and datasets

- Beware files in z/OS UNIX are *byte oriented,* text files are terminated by a "newline" character, X'15'
  - MVS datasets are record oriented, there is no newline, each record (line) has its length. Depending on the record format the length of lines may be fixed or vary. In fixed format datasets text is usually padded with blanks
  - You have to know if you want to replace newlines with trailing spaces or not (and vice versa) when copying back and forth

```
> od -Ax -c -tx1 test
0000000000     H   e   l   l   o   !       ^   [   ]      \n   W   h   a   t
              C8  85  93  93  96  5A  40  5F  AD  BD  40  15  E6  88  81  A3
0000000010     '   s       u   p   .     260 272 273  \n
              7D  A2  40  A4  97  4B  40  B0  BA  BB  15
000000001B
> 
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

Unix files are just a stream of bytes without an internal structure. It is the program who interprets the contents, not the file system. The picture shows a dump of a file called "test" produced by an "od" (Octal Dump) command. The "\n" symbol is the standard way of referring to the newline character.



Copying data between files and datasets

- Beware z/OS UNIX files use IBM-1047 as default encoding
  - MVS datasets use IBM-037 as default, these differ in ^ [ ]
  - When copying between files and datasets you have to know if you want to convert the encoding or not (e.g. C source code)

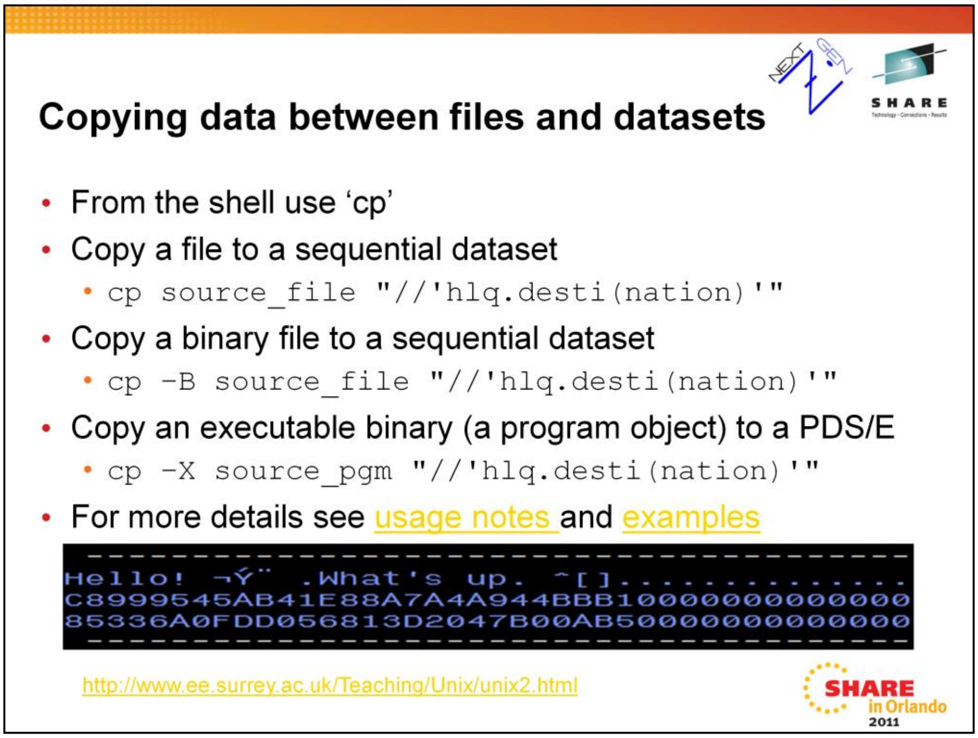
```
 ---------------------------------------
Hello! ¬Ý¨
C8999545AB444444444444444444444444444444
85336A0FDD000000000000000000000000000000
 ---------------------------------------
What's up. ^[]
E88A7A4A944BBB44444444444444444444444444
6813D2047B00AB00000000000000000000000000
 ---------------------------------------
********************************* Bottom
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

SHARE in Orlando 2011

The picture on this slide shows a (text) copy of the file test into a dataset with fixed record length. Notice that newline characters have been removed and replaced with blank (X'40') characters padding each line to it's full length.

Also notice that even though the bytes X'5F', X'AD', X'BD', X'B0', X'BA', X'BB' haven't physically changed they are displayed as different characters. This is due to the setting of the terminal to display data as IBM-037 on this (3270) terminal, while to IBM-1047 in the unix terminal on previous slide.



Copying data between files and datasets

- From the shell use ‘cp’
- Copy a file to a sequential dataset
  - `cp source_file "//'hlq.desti(nation)'"`
- Copy a binary file to a sequential dataset
  - `cp -B source_file "//'hlq.desti(nation)'"`
- Copy an executable binary (a program object) to a PDS/E
  - `cp -X source_pgm "//'hlq.desti(nation)'"`
- For more details see usage notes and examples

```
 ---------------------------------------
Hello! ¬Ý¨ .What's up. ^[]..............
C8999545AB41E88A7A4A944BBB10000000000000
85336A0FDD056813D2047B00AB50000000000000
 ---------------------------------------
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

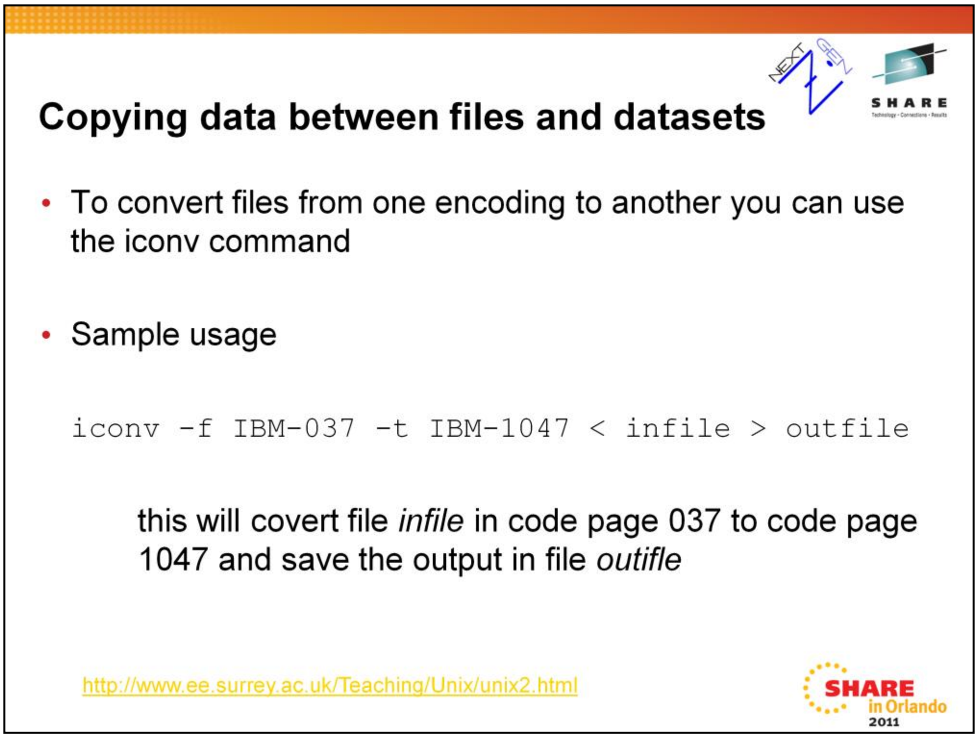
SHARE in Orlando 2011

The extra quotes around `//'hlq.desti(nation)'` are present because they tell the shell this is a single parameter to be passed to the program 'cp'. The shell removes them before passing the argument to the program.

Copying "program objects" from a file system to a PDS can result in unpredictable results since the conversion to a "load module" may or may not be successful. Use PDS/E instead.

The picture on this slide shows the result of copying the very same file "test" into a dataset having a fixed record length but this time in a binary mode via cp with the -B option. This time we got a single record including X'15' (newline) characters and the record is padded with binary zeros to it's full length.

If truncation is to occur (record length smaller than a line length) cp fails not copying anything and issuing an error message.



Copying data between files and datasets

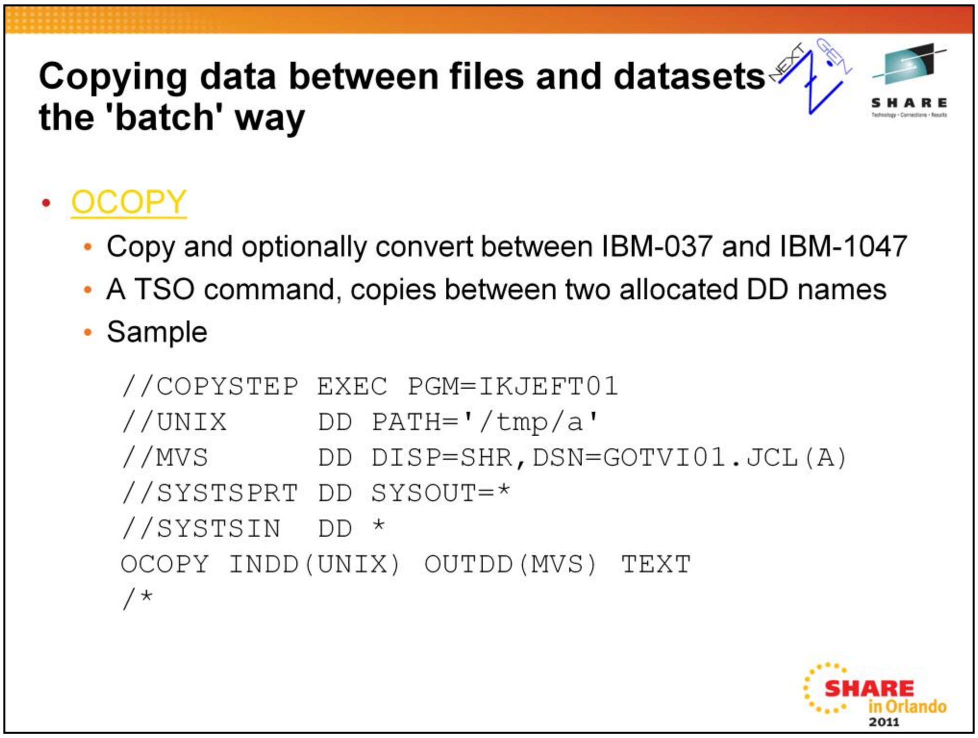
- To convert files from one encoding to another you can use the iconv command
- Sample usage

```
iconv -f IBM-037 -t IBM-1047 < infile > outfile
```

this will covert file *infile* in code page 037 to code page 1047 and save the output in file *outifle*

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

In case you are interested in the history of IBM 1047, this is an interesting discussion http://groups.google.com/group/bit.listserv.ibm-main/msg/0b633f8476306da0 it seems to relate to SHARE.



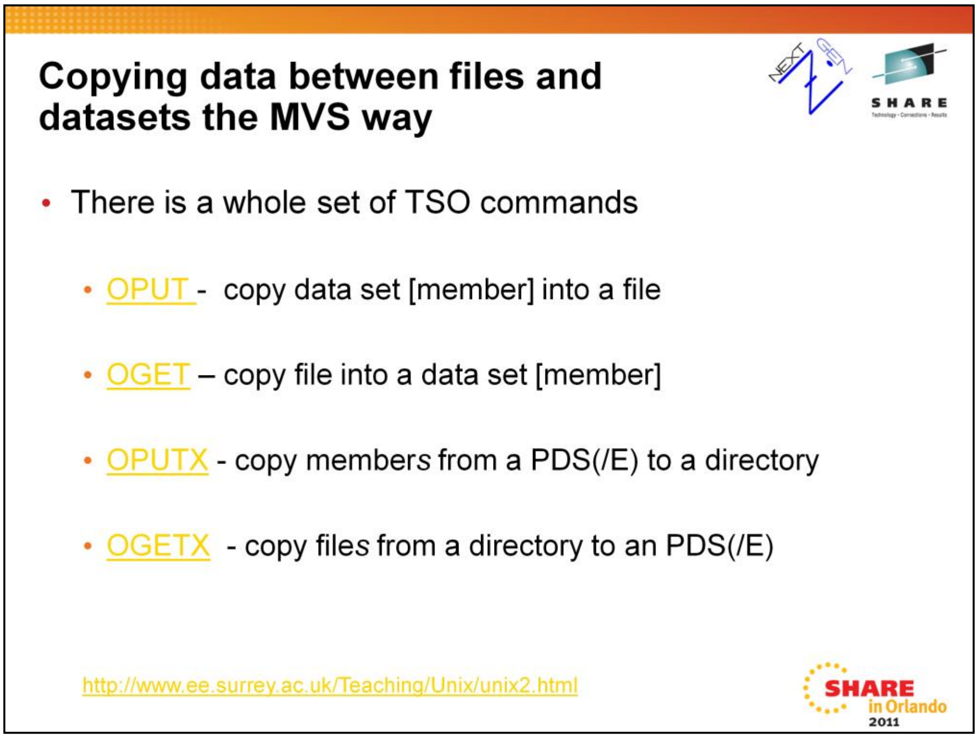
Copying data between files and datasets the 'batch' way

- OCOPY
  - Copy and optionally convert between IBM-037 and IBM-1047
  - A TSO command, copies between two allocated DD names
  - Sample

```
//COPYSTEP EXEC PGM=IKJEFT01
//UNIX     DD PATH='/tmp/a'
//MVS      DD DISP=SHR,DSN=GOTVI01.JCL(A)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
OCOPY INDD(UNIX) OUTDD(MVS) TEXT
/*
```

SHARE in Orlando 2011

There is also a BPXCOPY utility. However I never used it.

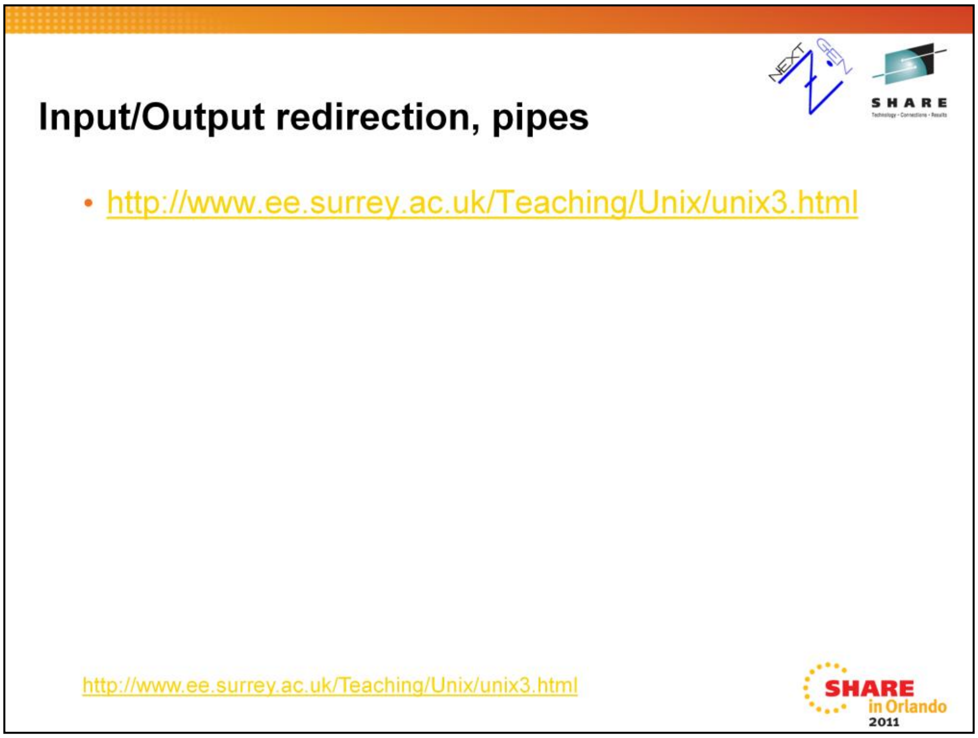


Copying data between files and datasets the MVS way

- There is a whole set of TSO commands
  - OPUT - copy data set [member] into a file
  - OGET – copy file into a data set [member]
  - OPUTX - copy members from a PDS(/E) to a directory
  - OGETX - copy files from a directory to an PDS(/E)

http://www.ee.surrey.ac.uk/Teaching/Unix/unix2.html

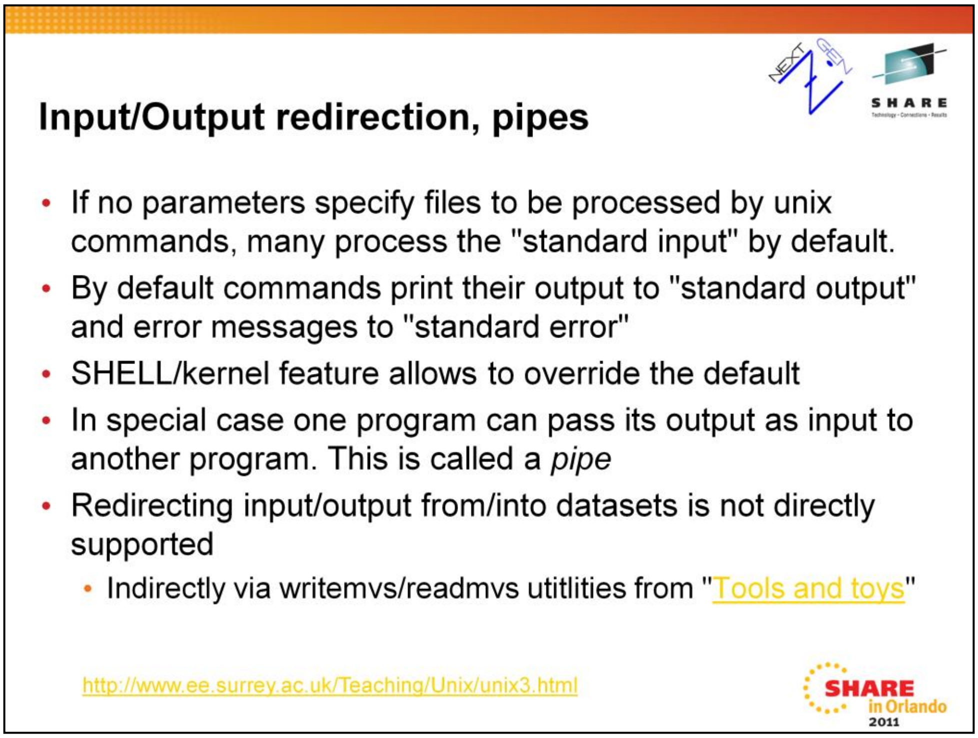
See the manual (click on the particular command) to find out more about these and programs.



Input/Output redirection, pipes

- http://www.ee.surrey.ac.uk/Teaching/Unix/unix3.html

http://www.ee.surrey.ac.uk/Teaching/Unix/unix3.html



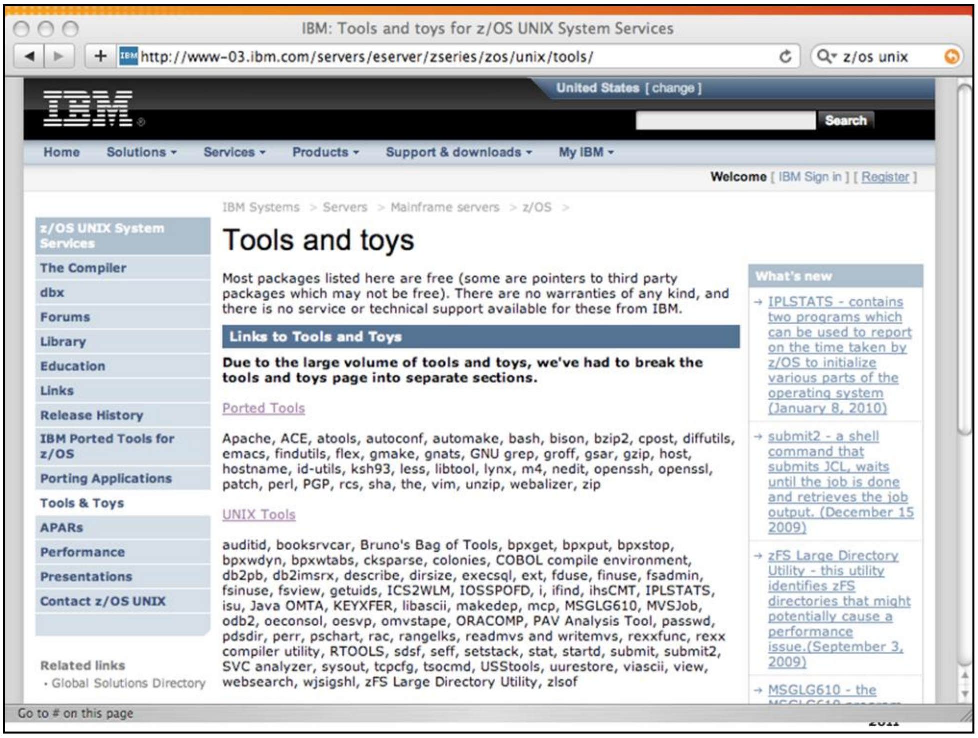
Input/Output redirection, pipes

- If no parameters specify files to be processed by unix commands, many process the "standard input" by default.
- By default commands print their output to "standard output" and error messages to "standard error"
- SHELL/kernel feature allows to override the default
- In special case one program can pass its output as input to another program. This is called a *pipe*
- Redirecting input/output from/into datasets is not directly supported
  - Indirectly via writemvs/readmvs utitlities from "Tools and toys"

http://www.ee.surrey.ac.uk/Teaching/Unix/unix3.html

When executing commands in z/OS UNIX shell, we do not have a JCL available to specify the input and output as we do in batch. To provide for this, each command being executed has a predefined "standard input", "standard output", and "standard error" to read input and write output and errors. Standard input is typically mapped to the input from keyboard, standard output and standard error to the terminal screen.

Special shell syntax makes it possible to redefine the standard input/output/error and let the commands to read/output whatever the user asked for.


IBM: Tools and toys for z/OS UNIX System Services
http://www-03.ibm.com/servers/eserver/zseries/zos/unix/tools/
z/os unix
IBM
United States [change]
Search
Home
Solutions
Services
Products
Support & downloads
My IBM
Welcome [IBM Sign in] [Register]
IBM Systems > Servers > Mainframe servers > z/OS >
z/OS UNIX System Services
The Compiler
dbx
Forums
Library
Education
Links
Release History
IBM Ported Tools for z/OS
Porting Applications
Tools & Toys
APARs
Performance
Presentations
Contact z/OS UNIX
Related links
· Global Solutions Directory
Go to # on this page

Tools and toys

Most packages listed here are free (some are pointers to third party packages which may not be free). There are no warranties of any kind, and there is no service or technical support available for these from IBM.

Links to Tools and Toys

Due to the large volume of tools and toys, we've had to break the tools and toys page into separate sections.

Ported Tools

Apache, ACE, atools, autoconf, automake, bash, bison, bzip2, cpost, diffutils, emacs, findutils, flex, gmake, gnats, GNU grep, groff, gsar, gzip, host, hostname, id-utils, ksh93, less, libtool, lynx, m4, nedit, openssh, openssl, patch, perl, PGP, rcs, sha, the, vim, unzip, webalizer, zip

UNIX Tools

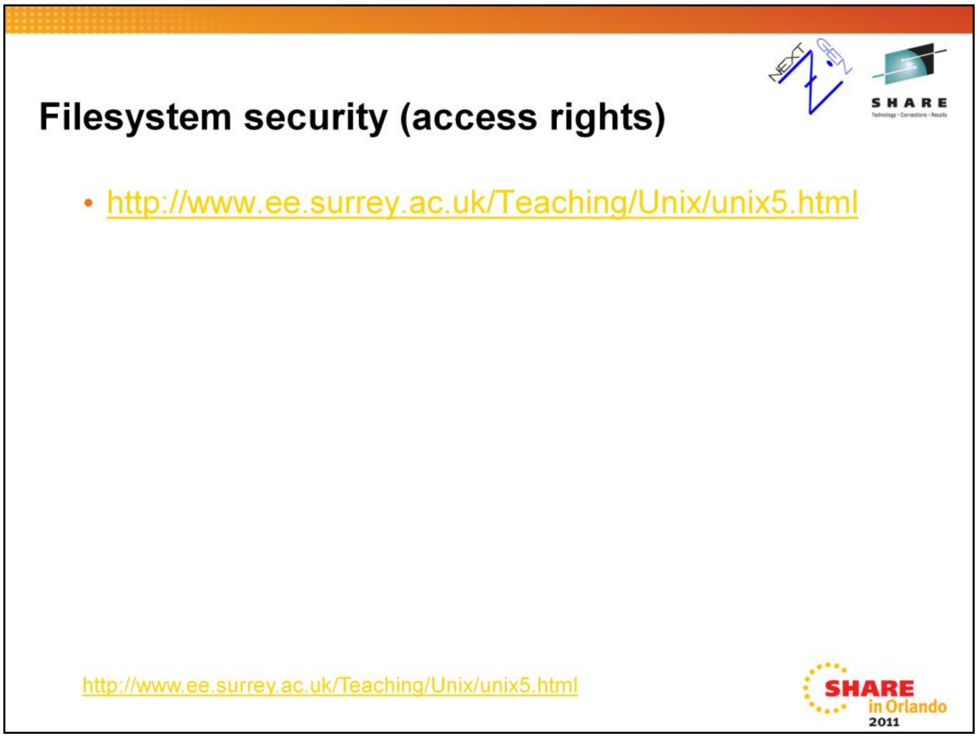
auditid, booksrvcar, Bruno's Bag of Tools, bpxget, bpxput, bpxstop, bpxwdyn, bpxwtabs, cksparse, colonies, COBOL compile environment, db2pb, db2imsrx, describe, dirsize, execsql, ext, fduse, finuse, fsadmin, fsinuse, fsview, getuids, ICS2WLM, IOSSPOFD, i, ifind, ihsCMT, IPLSTATS, isu, Java OMTA, KEYXFER, libascii, makedep, mcp, MSGLG610, MVSJob, odb2, oeconsol, oesvp, omvstape, ORACOMP, PAV Analysis Tool, passwd, pdsdir, perr, pschart, rac, rangelks, readmvs and writemvs, rexxfunc, rexx compiler utility, RTOOLS, sdsf, seff, setstack, stat, startd, submit, submit2, SVC analyzer, sysout, tcpcfg, tsocmd, USStools, uurestore, viascii, view, websearch, wjsigshl, zFS Large Directory Utility, zlsof

What's new

- → IPLSTATS - contains two programs which can be used to report on the time taken by z/OS to initialize various parts of the operating system (January 8, 2010)
- → submit2 - a shell command that submits JCL, waits until the job is done and retrieves the job output. (December 15 2009)
- → zFS Large Directory Utility - this utility identifies zFS directories that might potentially cause a performance issue.(September 3, 2009)
- → MSGLG610 - the

Several MVS specific utilities are included. If you want to list PDS members from a shell, there is a ‘pdsdir’ utitlity. If you want to pipe from/to data set there are ‘readmvs’ / ‘writemvs’ utitlities.

There are also many useful tools including BASH and several GNU utilities, most notably ‘GNU grep’ which also supports processing of datasets (even groups of datasets by specifying a recursive mode via –r and providing a generic DSN pattern like in ISPF ‘Data Set List Utility’ (i.e. in the form that Catalog Search Interface understands).



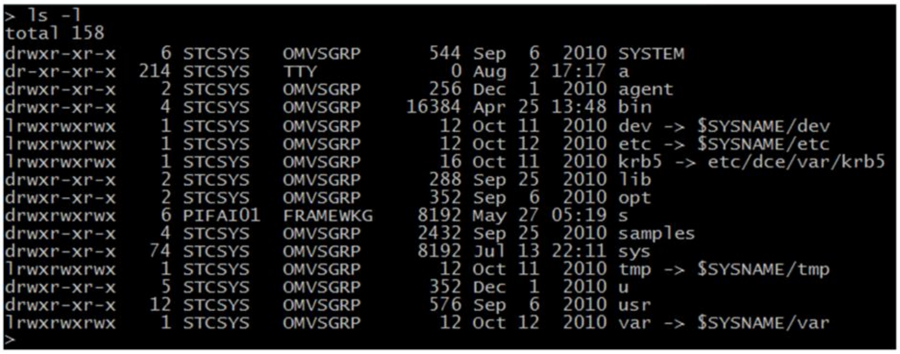
Filesystem security (access rights)

- http://www.ee.surrey.ac.uk/Teaching/Unix/unix5.html

http://www.ee.surrey.ac.uk/Teaching/Unix/unix5.html

Filesystem security (access rights)

- Command `ls -l` displays many file attributes



```
> ls -l
total 158
drwxr-xr-x    6 STCSYS   OMVSGRP      544 Sep  6  2010 SYSTEM
dr-xr-xr-x  214 STCSYS   TTY            0 Aug  2 17:17 a
drwxr-xr-x    2 STCSYS   OMVSGRP      256 Dec  1  2010 agent
drwxr-xr-x    4 STCSYS   OMVSGRP    16384 Apr 25 13:48 bin
lrwxrwxrwx    1 STCSYS   OMVSGRP       12 Oct 11  2010 dev -> $SYSNAME/dev
lrwxrwxrwx    1 STCSYS   OMVSGRP       12 Oct 12  2010 etc -> $SYSNAME/etc
lrwxrwxrwx    1 STCSYS   OMVSGRP       16 Oct 11  2010 krb5 -> etc/dce/var/krb5
drwxr-xr-x    2 STCSYS   OMVSGRP      288 Sep 25  2010 lib
drwxr-xr-x    2 STCSYS   OMVSGRP      352 Sep  6  2010 opt
drwxrwxrwx    6 PIFAI01  FRAMEWKG    8192 May 27 05:19 s
drwxr-xr-x    4 STCSYS   OMVSGRP     2432 Sep 25  2010 samples
drwxr-xr-x   74 STCSYS   OMVSGRP     8192 Jul 13 22:11 sys
lrwxrwxrwx    1 STCSYS   OMVSGRP       12 Oct 11  2010 tmp -> $SYSNAME/tmp
drwxr-xr-x    5 STCSYS   OMVSGRP      352 Dec  1  2010 u
drwxr-xr-x   12 STCSYS   OMVSGRP      576 Sep  6  2010 usr
lrwxrwxrwx    1 STCSYS   OMVSGRP       12 Oct 12  2010 var -> $SYSNAME/var
>
```

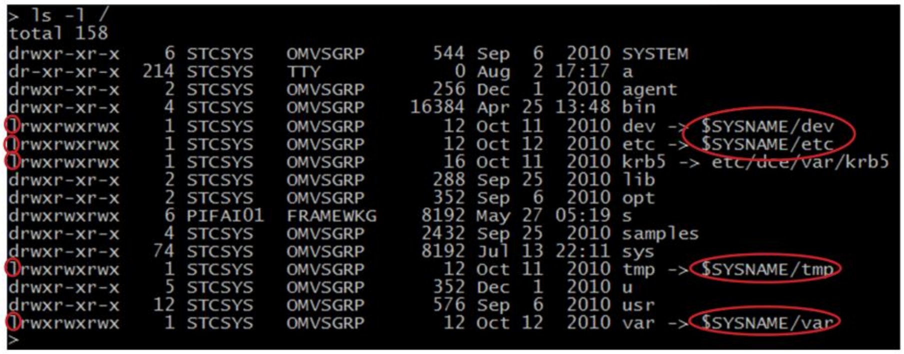
http://www.ee.surrey.ac.uk/Teaching/Unix/unix5.html

SHARE in Orlando 2011

The ls command also has the -E switch that prints all that –l does, but provides some more z/OS UNIX specific attributes like APF authorization or if a program library is shared or not.

Filesystem security (access rights)

- Command `ls -l` displays among others *symbolic links*
- There is session 9875: z/OS UNIX Shared FileSystem ...



```
> ls -l /
total 158
drwxr-xr-x    6 STCSYS    OMVSGRP       544 Sep  6  2010 SYSTEM
dr-xr-xr-x  214 STCSYS    TTY             0 Aug  2 17:17 a
drwxr-xr-x    2 STCSYS    OMVSGRP       256 Dec  1  2010 agent
drwxr-xr-x    4 STCSYS    OMVSGRP     16384 Apr 25 13:48 bin
lrwxrwxrwx    1 STCSYS    OMVSGRP        12 Oct 11  2010 dev -> $SYSNAME/dev
lrwxrwxrwx    1 STCSYS    OMVSGRP        12 Oct 12  2010 etc -> $SYSNAME/etc
lrwxrwxrwx    1 STCSYS    OMVSGRP        16 Oct 11  2010 krb5 -> etc/dce/var/krb5
drwxr-xr-x    2 STCSYS    OMVSGRP       288 Sep 25  2010 lib
drwxr-xr-x    2 STCSYS    OMVSGRP       352 Sep  6  2010 opt
drwxrwxrwx    6 PIFAI01   FRAMEWKG     8192 May 27 05:19 s
drwxr-xr-x    4 STCSYS    OMVSGRP      2432 Sep 25  2010 samples
drwxr-xr-x   74 STCSYS    OMVSGRP      8192 Jul 13 22:11 sys
lrwxrwxrwx    1 STCSYS    OMVSGRP        12 Oct 11  2010 tmp -> $SYSNAME/tmp
drwxr-xr-x    5 STCSYS    OMVSGRP       352 Dec  1  2010 u
drwxr-xr-x   12 STCSYS    OMVSGRP       576 Sep  6  2010 usr
lrwxrwxrwx    1 STCSYS    OMVSGRP        12 Oct 12  2010 var -> $SYSNAME/var
>
```

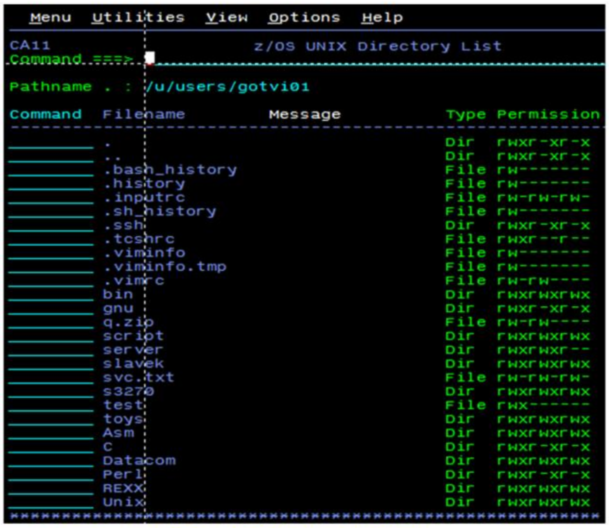
http://www.ee.surrey.ac.uk/Teaching/Unix/unix5.html

Files for which the output of ls -l starts with an "l" are so called "symbolic links"

In z/OS UNIX there is a special type of a symbolic link whose name starts with a $SYSNAME/… This refers to the symbolic &SYSNAME as defined in SYS1.PARMLIB for a SYSPLEX environement. If SYSPLEX is specified as NO in SYS1.PARMLIB(BPXPRMxx), then $SYSNAME is replaced with /SYSTEM when the symbolic link is resolved.

For more see http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.bpxb200/filesing.htm

Filesystem security (access rights), OBROWSE

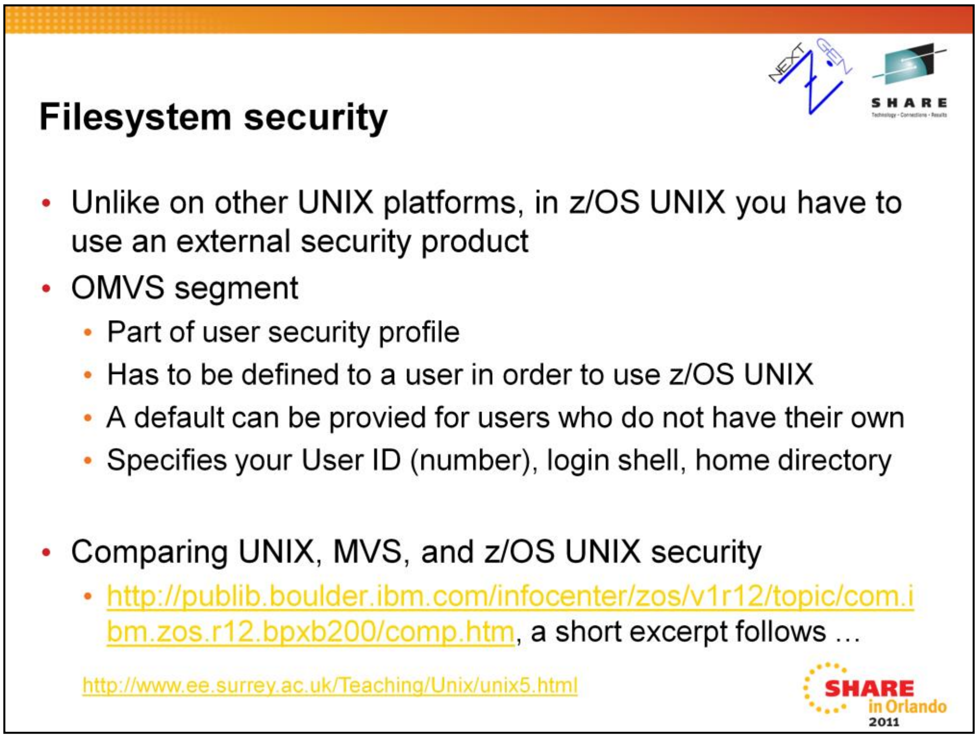


```
  Menu  Utilities  View  Options  Help
CA11                           z/OS UNIX Directory List
Command ===>
Pathname . : /u/users/gotvi01
Command  Filename         Message          Type Permission
-------------------------------------------------------------
_________ .                                Dir  rwxr-xr-x
_________ ..                               Dir  rwxr-xr-x
_________ .bash_history                    File rw-------
_________ .history                         File rw-------
_________ .inputrc                         File rw-rw-rw-
_________ .sh_history                      File rw-------
_________ .ssh                             Dir  rwxr-xr-x
_________ .tcshrc                          File rwxr--r--
_________ .viminfo                         File rw-------
_________ .viminfo.tmp                     File rw-------
_________ .vimrc                           File rw-rw----
_________ bin                              Dir  rwxrwxrwx
_________ gnu                              Dir  rwxr-xr-x
_________ q.zip                            File rw-rw----
_________ script                           Dir  rwxrwxrwx
_________ server                           Dir  rwxrwxr--
_________ slavek                           Dir  rwxrwxrwx
_________ svc.txt                          File rw-rw-rw-
_________ s3270                            Dir  rwxrwxrwx
_________ test                             File rwx------
_________ toys                             Dir  rwxrwxrwx
_________ Asm                              Dir  rwxrwxrwx
_________ C                                Dir  rwxr-xr-x
_________ Datacom                          Dir  rwxrwxrwx
_________ Perl                             Dir  rwxr-xr-x
_________ REXX                             Dir  rwxrwxrwx
_________ Unix                             Dir  rwxrwxrwx
*************************************************************
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix5.html

SHARE in Orlando 2011

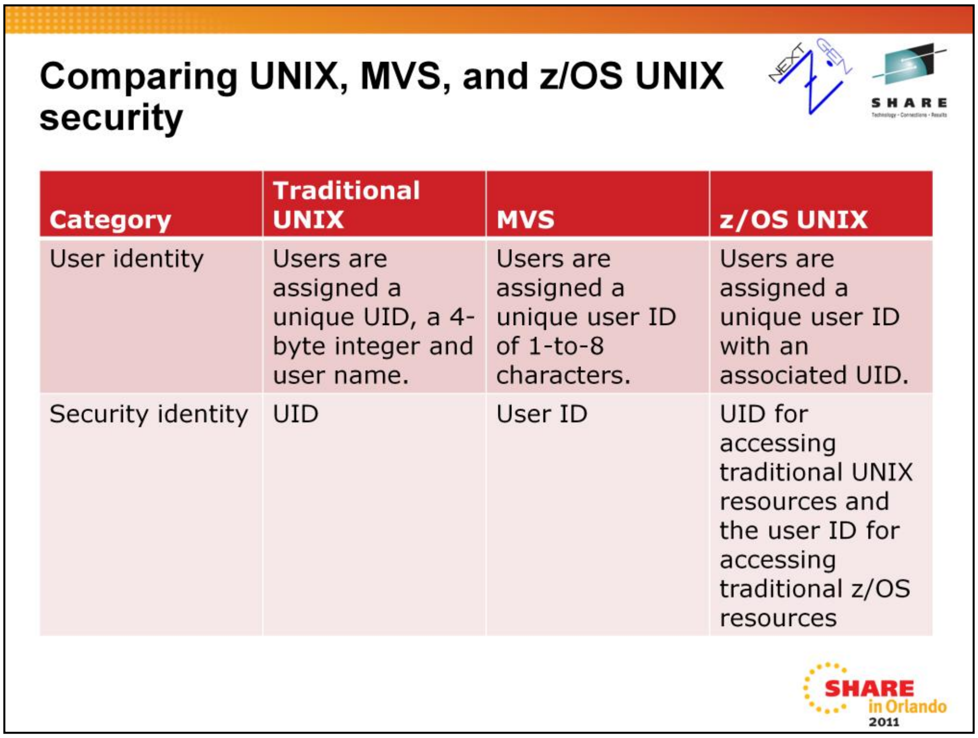
From ISPF or OMVS we can use OBROWSE (or ISPF P.3.2) to show output similar to `ls -l`



Filesystem security

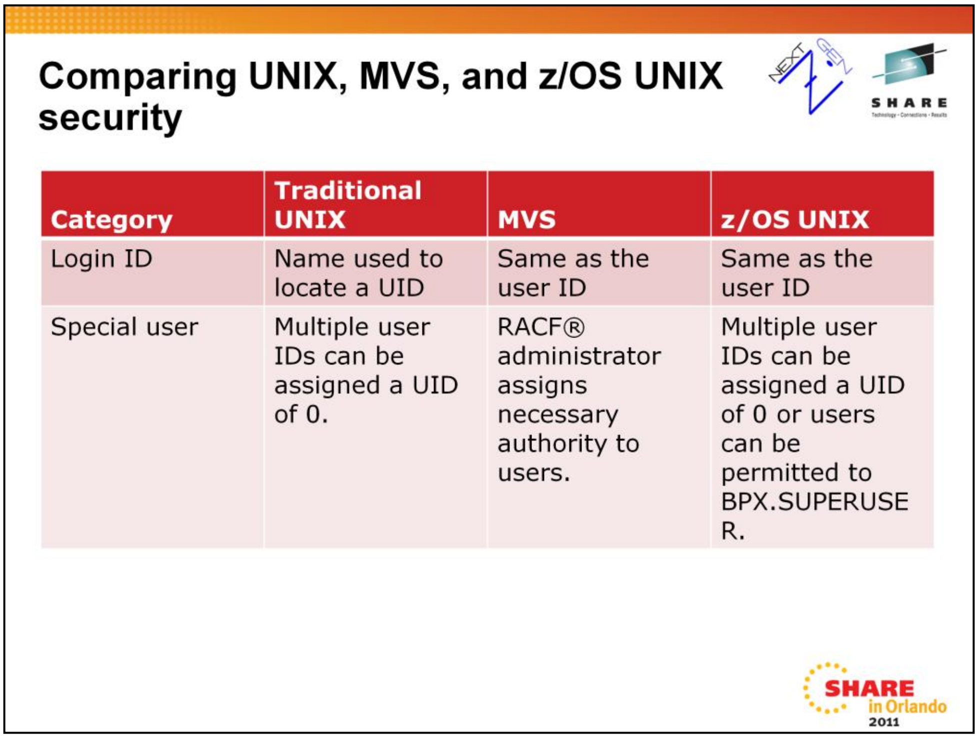
- Unlike on other UNIX platforms, in z/OS UNIX you have to use an external security product
- OMVS segment
  - Part of user security profile
  - Has to be defined to a user in order to use z/OS UNIX
  - A default can be provied for users who do not have their own
  - Specifies your User ID (number), login shell, home directory
- Comparing UNIX, MVS, and z/OS UNIX security
  - http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.bpxb200/comp.htm, a short excerpt follows …

http://www.ee.surrey.ac.uk/Teaching/Unix/unix5.html



Comparing UNIX, MVS, and z/OS UNIX security

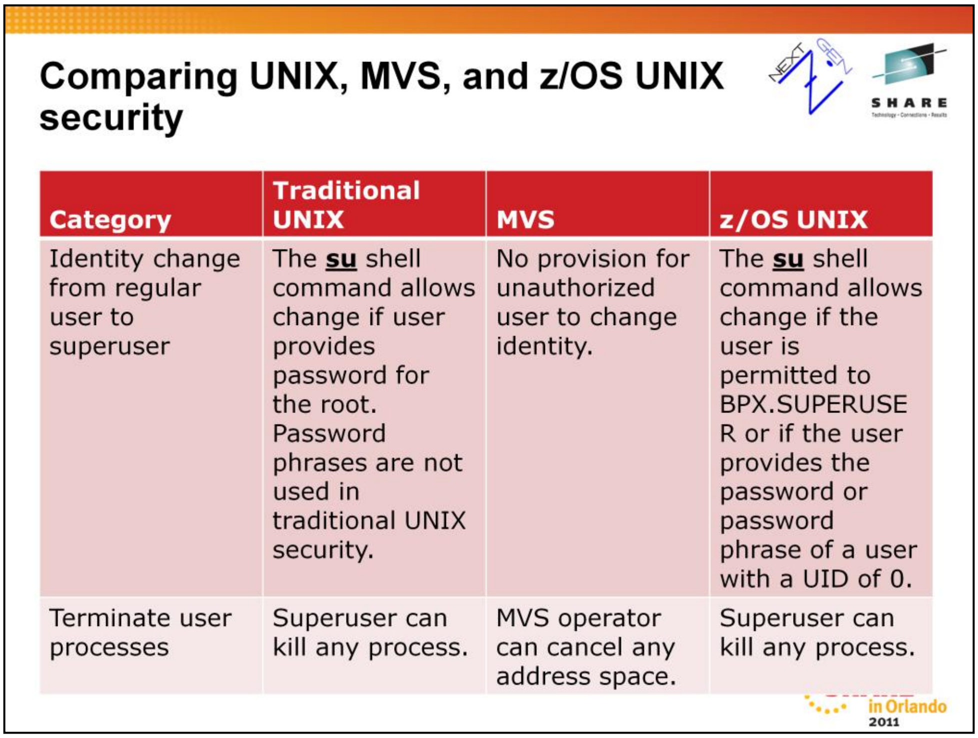
Category	Traditional UNIX	MVS	z/OS UNIX
User identity	Users are assigned a unique UID, a 4-byte integer and user name.	Users are assigned a unique user ID of 1-to-8 characters.	Users are assigned a unique user ID with an associated UID.
Security identity	UID	User ID	UID for accessing traditional UNIX resources and the user ID for accessing traditional z/OS resources



Comparing UNIX, MVS, and z/OS UNIX security

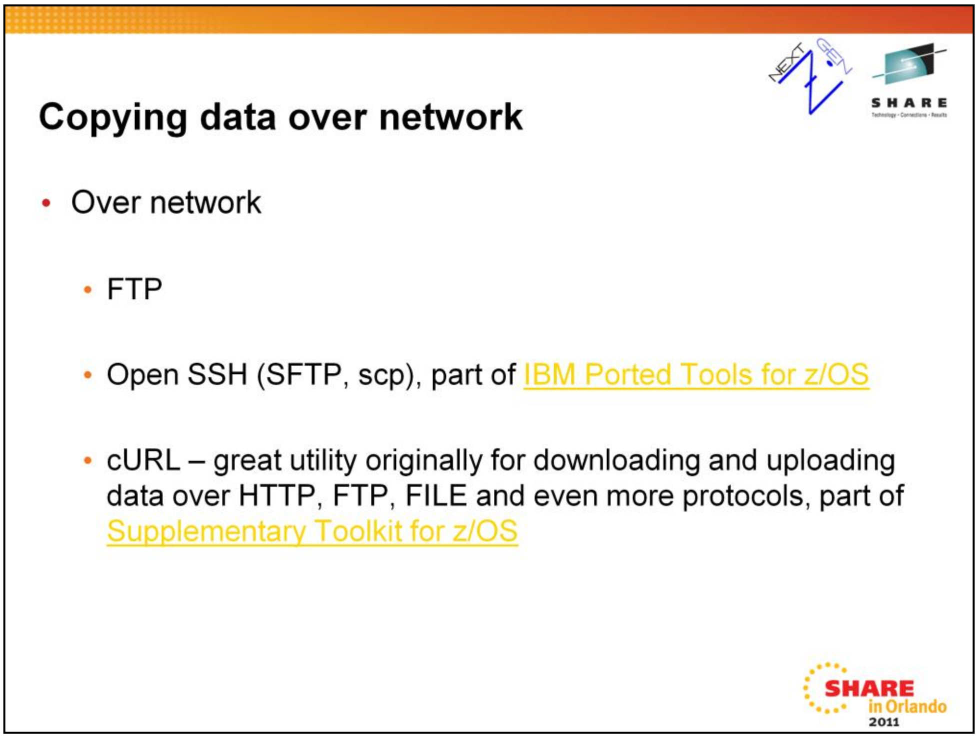
Category	Traditional UNIX	MVS	z/OS UNIX
Login ID	Name used to locate a UID	Same as the user ID	Same as the user ID
Special user	Multiple user IDs can be assigned a UID of 0.	RACF® administrator assigns necessary authority to users.	Multiple user IDs can be assigned a UID of 0 or users can be permitted to BPX.SUPERUSER.

NEXT GEN
SHARE
Technology · Connections · Results
SHARE in Orlando 2011



Comparing UNIX, MVS, and z/OS UNIX security

Category	Traditional UNIX	MVS	z/OS UNIX
Identity change from regular user to superuser	The **su** shell command allows change if user provides password for the root. Password phrases are not used in traditional UNIX security.	No provision for unauthorized user to change identity.	The **su** shell command allows change if the user is permitted to BPX.SUPERUSER or if the user provides the password or password phrase of a user with a UID of 0.
Terminate user processes	Superuser can kill any process.	MVS operator can cancel any address space.	Superuser can kill any process.

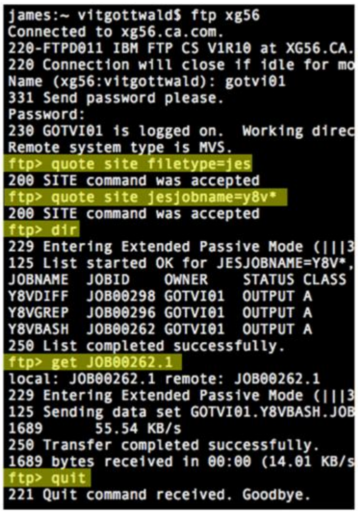


Copying data over network

- Over network
  - FTP
  - Open SSH (SFTP, scp), part of IBM Ported Tools for z/OS
  - cURL – great utility originally for downloading and uploading data over HTTP, FTP, FILE and even more protocols, part of Supplementary Toolkit for z/OS

FTP

- remote access to
  - UNIX file system
    ```
    ascii
    get /u/users/gotvi01/a.txt
    ```
  - datasets
    ```
    ascii
    quote site ispfstats
    quote site sbdataconn=
              (ibm-037, iso8859-1)
    put ab.txt 'gotvi01.txt(ab)'
    ```
  - JES
    ```
    quote site filetype=jes
    get JOB1234.1
    ```
- `quote help site`



```
james:~ vitgottwald$ ftp xg56
Connected to xg56.ca.com.
220-FTPD011 IBM FTP CS V1R10 at XG56.CA.
220 Connection will close if idle for mo
Name (xg56:vitgottwald): gotvi01
331 Send password please.
Password:
230 GOTVI01 is logged on.  Working direc
Remote system type is MVS.
ftp> quote site filetype=jes
200 SITE command was accepted
ftp> quote site jesjobname=y8v*
200 SITE command was accepted
ftp> dir
229 Entering Extended Passive Mode (|||3
125 List started OK for JESJOBNAME=Y8V*,
JOBNAME  JOBID    OWNER    STATUS CLASS
Y8VDIFF  JOB00298 GOTVI01  OUTPUT A
Y8VGREP  JOB00296 GOTVI01  OUTPUT A
Y8VBASH  JOB00262 GOTVI01  OUTPUT A
250 List completed successfully.
ftp> get JOB00262.1
local: JOB00262.1 remote: JOB00262.1
229 Entering Extended Passive Mode (|||3
125 Sending data set GOTVI01.Y8VBASH.JOB
1689       55.54 KB/s
250 Transfer completed successfully.
1689 bytes received in 00:00 (14.01 KB/s
ftp> quit
221 Quit command received. Goodbye.
```

‘quote site ispfstats’ enables ISPF statistics processing when updating members of PDS or PDSE. It stores the date, time and user account name.

`‘quote site sbdataconn=(ibm-037, iso8859-1)’ sets the conversion table for file transfer in ascii (text) mode.` Beware that z/OS UNIX uses IBM-1047 as the default character encoding

while MVS uses IBM-037. These two encodings differ only in a couple of characters, among which are the ^ [ ] symbols. If you have these characters make sure to set the right encoding.

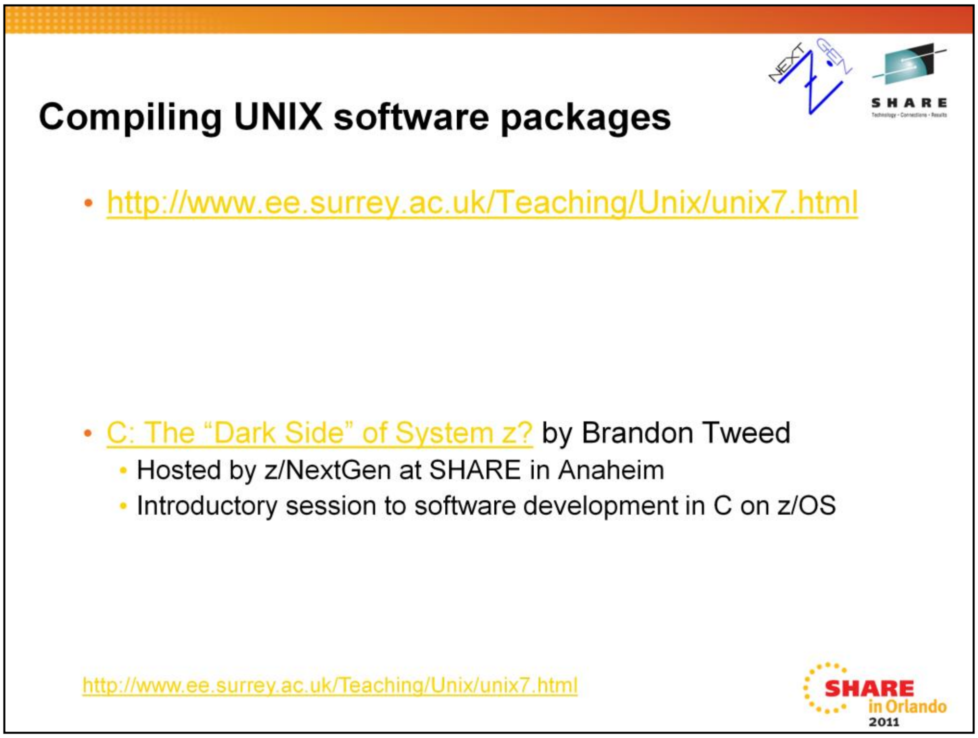
‘quote site filetype=jes’ switches the ftp client to JES mode and allows you to submit jobs (put) and download the output of other jobs (get).

‘quote help site’ shows the available site options with short help

For more information see

z/OS Communications Server IP User's Guide and Commands

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/F1A1B980/5.64?SHELF=F1A1BKA1&DT=20080602140814]



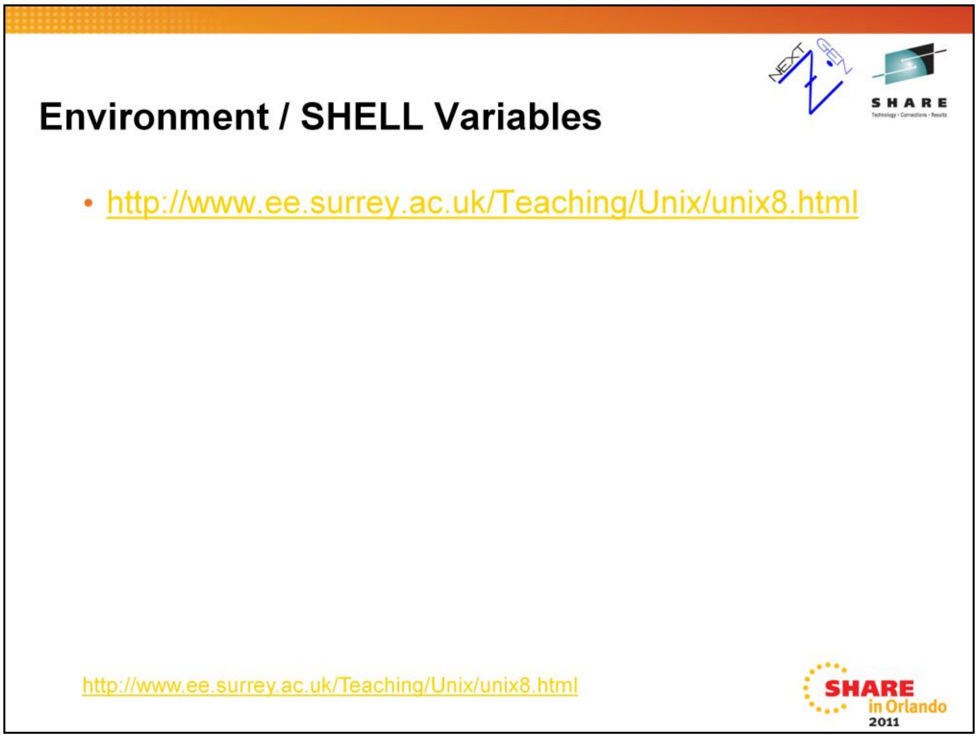
Compiling UNIX software packages

- http://www.ee.surrey.ac.uk/Teaching/Unix/unix7.html
- C: The "Dark Side" of System z? by Brandon Tweed
  - Hosted by z/NextGen at SHARE in Anaheim
  - Introductory session to software development in C on z/OS

http://www.ee.surrey.ac.uk/Teaching/Unix/unix7.html

SHARE in Orlando 2011

We do not have enough room in this session to go into details. Instead we reference a session from SHARE in Anaheim as a good point to get started.

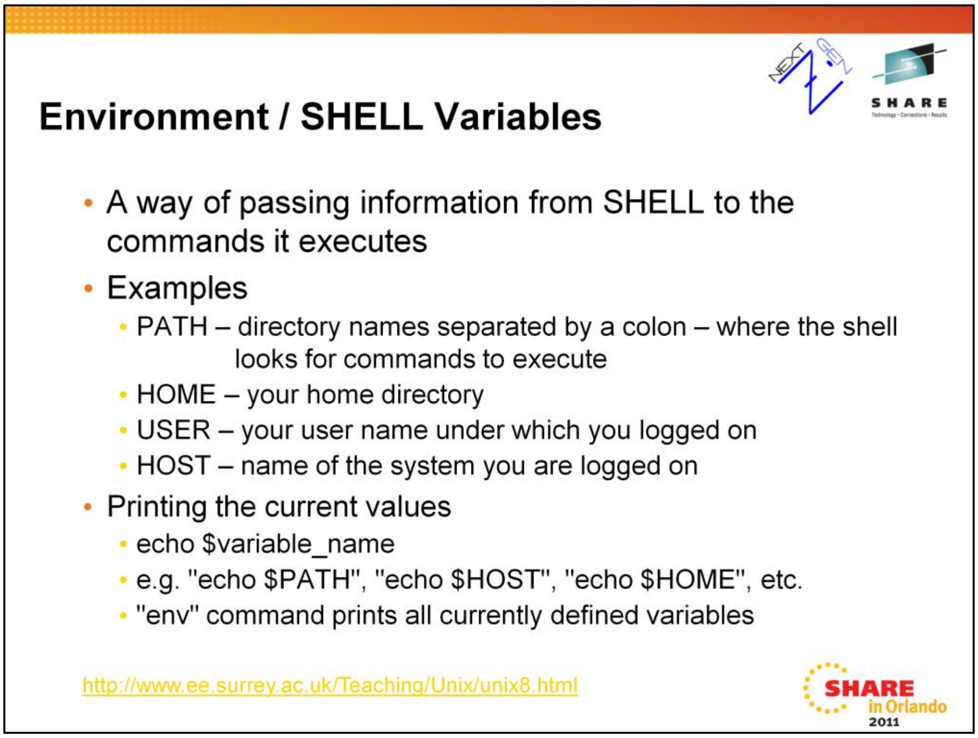


Environment / SHELL Variables

- http://www.ee.surrey.ac.uk/Teaching/Unix/unix8.html

http://www.ee.surrey.ac.uk/Teaching/Unix/unix8.html

SHARE in Orlando 2011

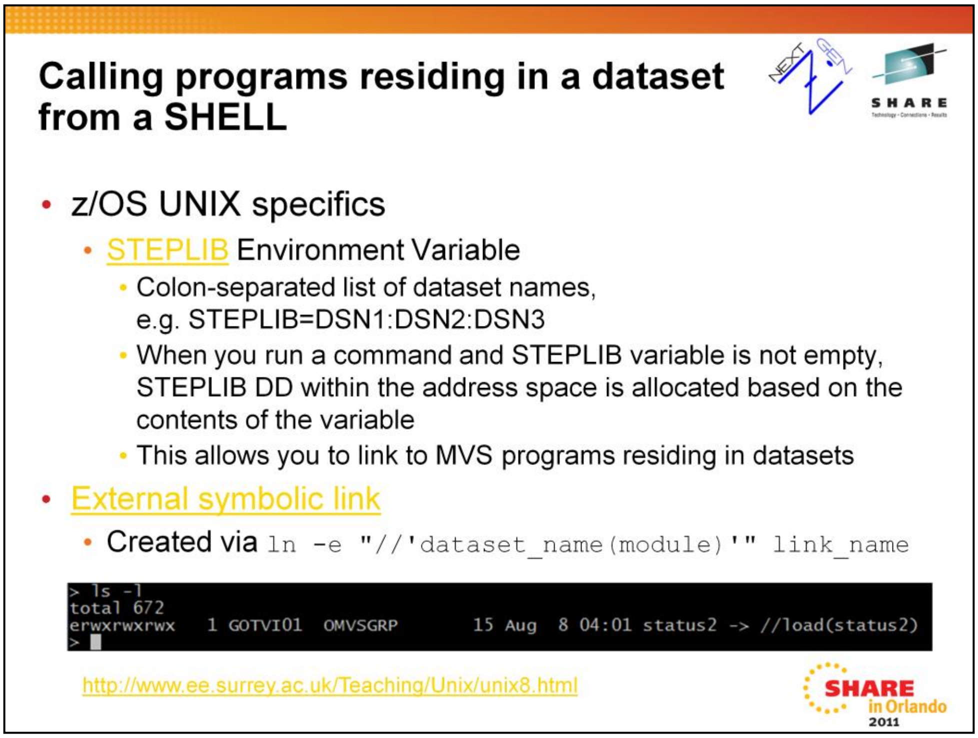


Environment / SHELL Variables

- A way of passing information from SHELL to the commands it executes
- Examples
  - PATH – directory names separated by a colon – where the shell looks for commands to execute
  - HOME – your home directory
  - USER – your user name under which you logged on
  - HOST – name of the system you are logged on
- Printing the current values
  - echo $variable_name
  - e.g. "echo $PATH", "echo $HOST", "echo $HOME", etc.
  - "env" command prints all currently defined variables

http://www.ee.surrey.ac.uk/Teaching/Unix/unix8.html

SHARE in Orlando 2011



Calling programs residing in a dataset from a SHELL

- z/OS UNIX specifics
  - STEPLIB Environment Variable
    - Colon-separated list of dataset names, e.g. STEPLIB=DSN1:DSN2:DSN3
    - When you run a command and STEPLIB variable is not empty, STEPLIB DD within the address space is allocated based on the contents of the variable
    - This allows you to link to MVS programs residing in datasets
- External symbolic link
  - Created via `ln -e "//'dataset_name(module)'" link_name`

```
> ls -l
total 672
erwxrwxrwx   1 GOTVI01   OMVSGRP       15 Aug  8 04:01 status2 -> //load(status2)
> 
```

http://www.ee.surrey.ac.uk/Teaching/Unix/unix8.html

SHARE in Orlando 2011

If you are an application programmer you should understand what parameter list your program receives when run it from a shell, from a JCL or from a TSO. Each of these environments, provides a different convention.

In case you are interested to see the difference in parameter passing, see:

Standard JCL PARM passing:

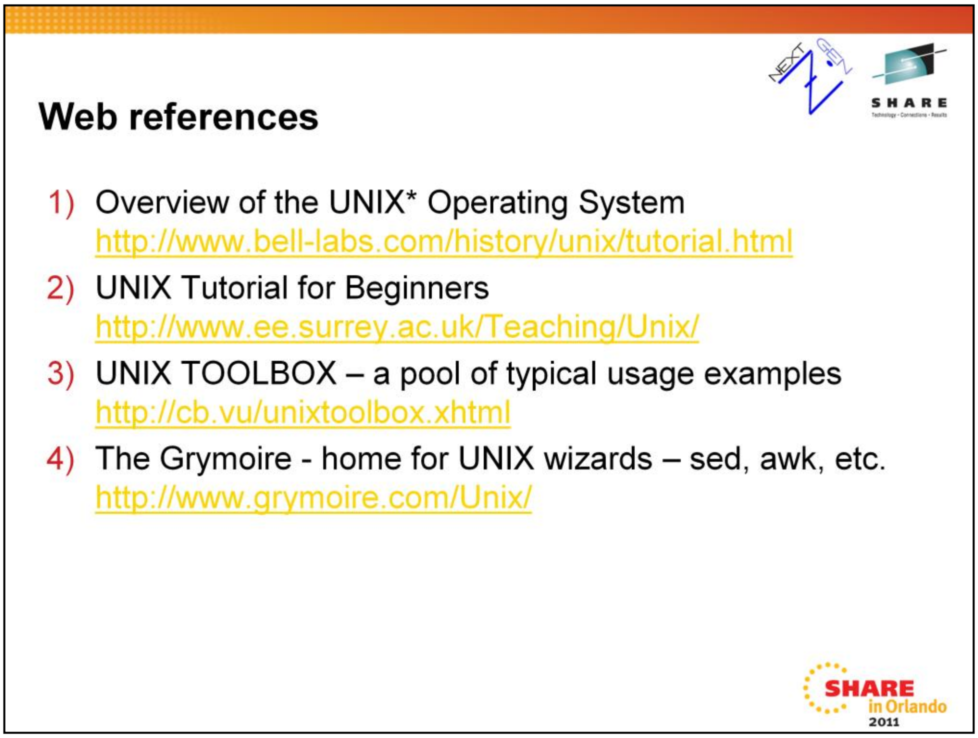
*
http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.ieaa600/parmpm.htm#parmpm

z/OS UNIX exec() parameter and environment passing:

*
http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.bpxb100/exc.htm#exc

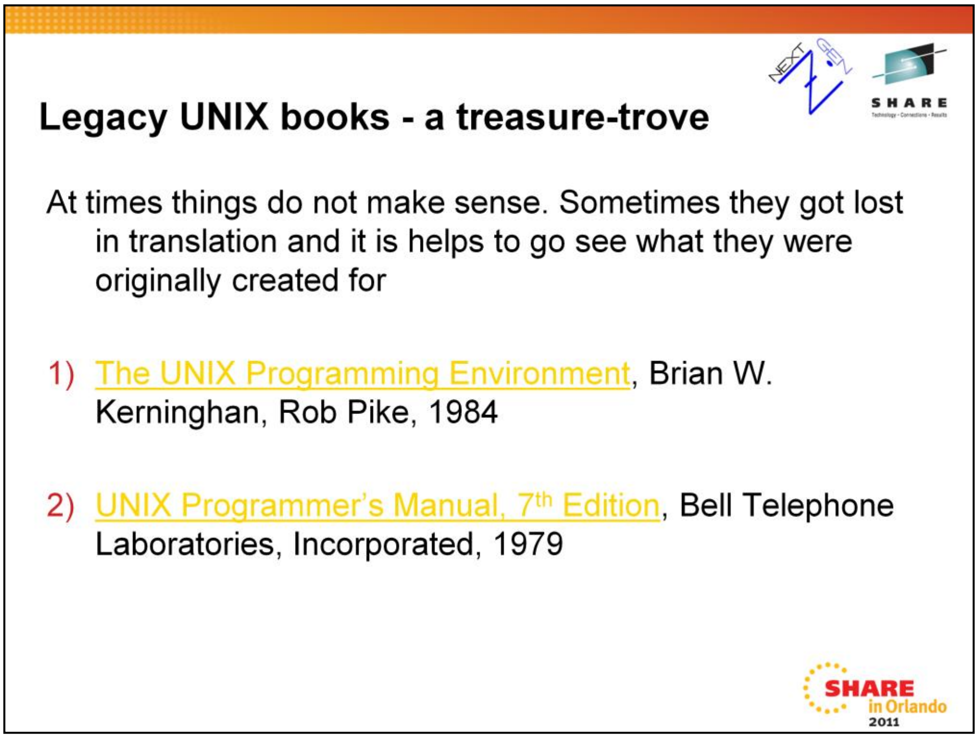
TSO Command Processor Parameter List (CPPL):

*
http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.ikjb600/ikj4b64011.htm#wq19



Web references

1) Overview of the UNIX* Operating System
   http://www.bell-labs.com/history/unix/tutorial.html
2) UNIX Tutorial for Beginners
   http://www.ee.surrey.ac.uk/Teaching/Unix/
3) UNIX TOOLBOX – a pool of typical usage examples
   http://cb.vu/unixtoolbox.xhtml
4) The Grymoire - home for UNIX wizards – sed, awk, etc.
   http://www.grymoire.com/Unix/



Legacy UNIX books - a treasure-trove

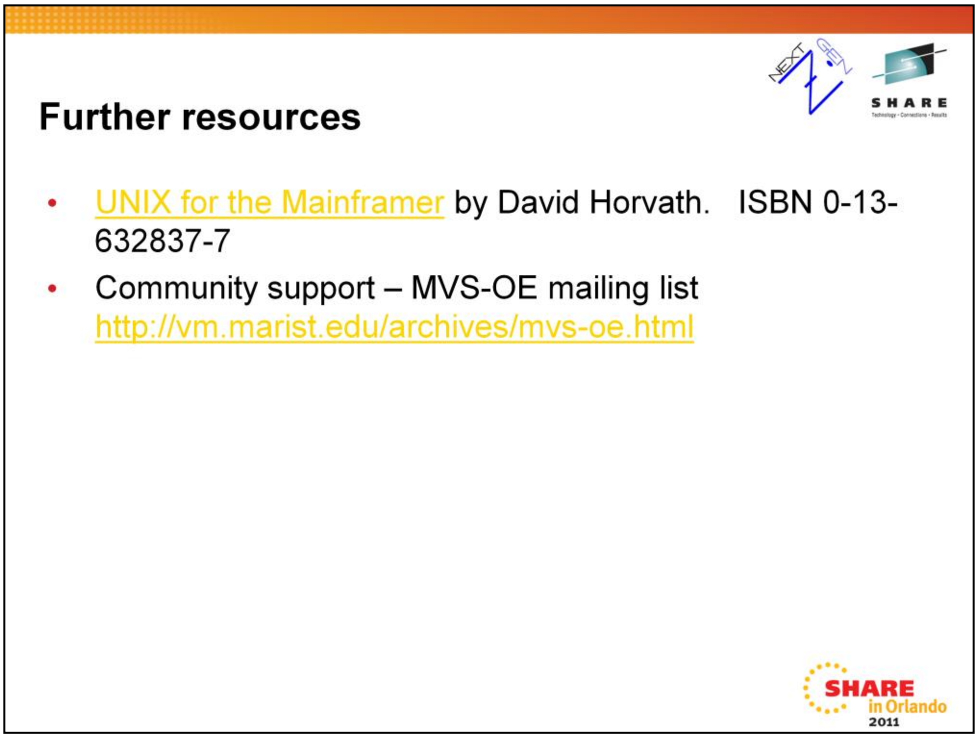
At times things do not make sense. Sometimes they got lost in translation and it is helps to go see what they were originally created for

1) The UNIX Programming Environment, Brian W. Kerninghan, Rob Pike, 1984

2) UNIX Programmer's Manual, 7th Edition, Bell Telephone Laboratories, Incorporated, 1979

I recommend reading 1). Things start to make sense when explained by original authors. First 170 pages are suitable for everyone learning the subject. The second half is for programmers. It has a great index at the end of the book!

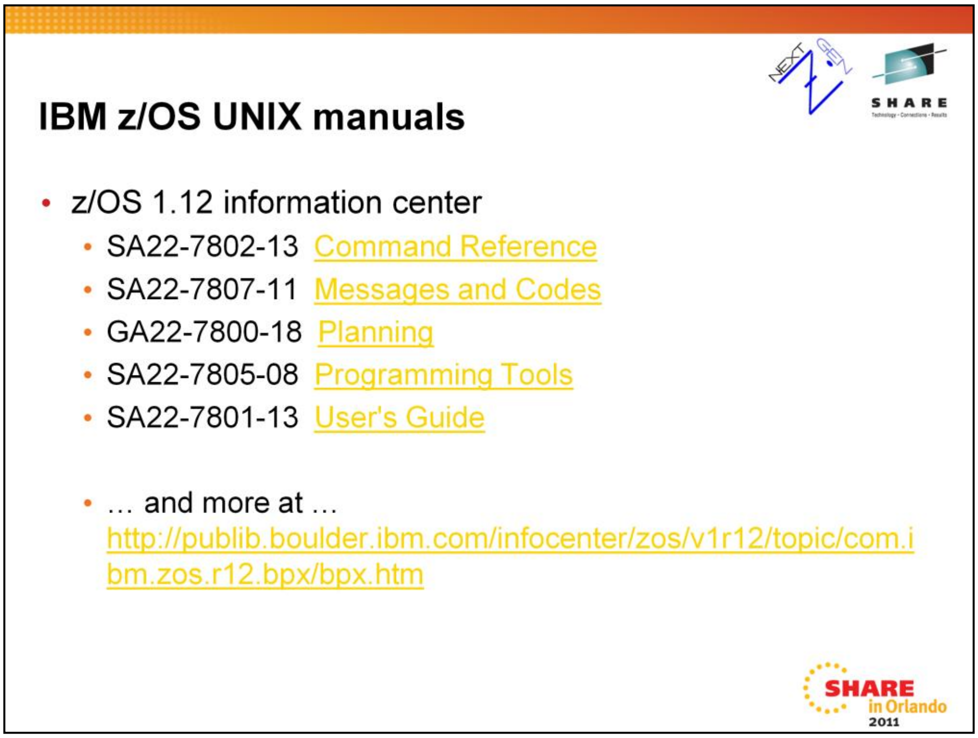
Consider the book 1) a guide and 2) a reference manual.



Further resources

- UNIX for the Mainframer by David Horvath.   ISBN 0-13-632837-7
- Community support – MVS-OE mailing list http://vm.marist.edu/archives/mvs-oe.html

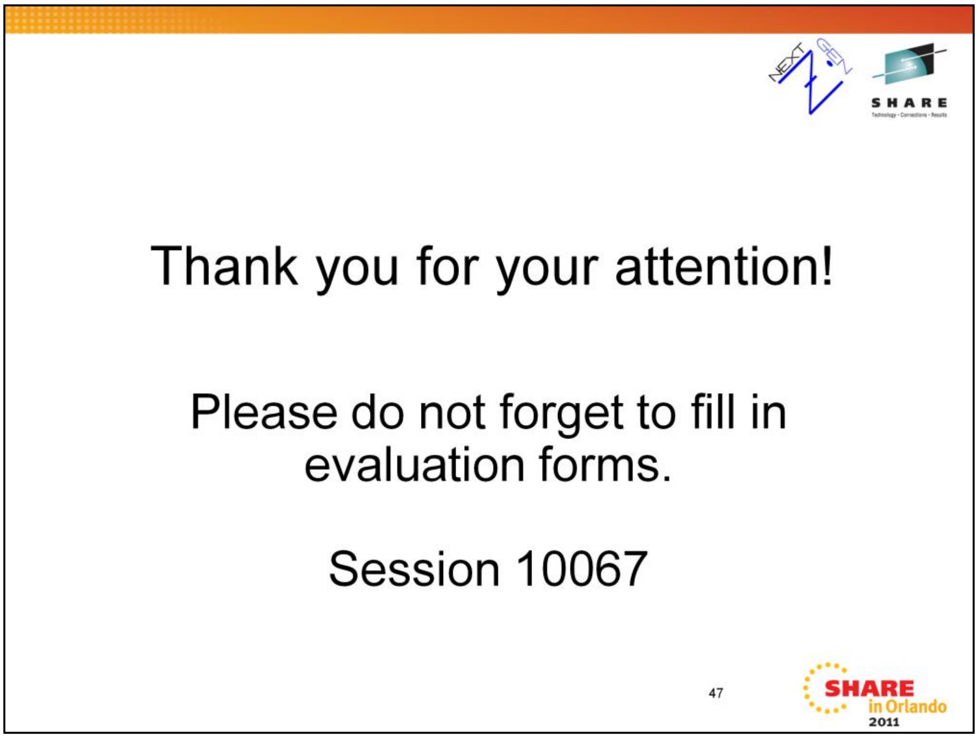
MVS-OE stands for MVS Open Edition, the original name of the product later called Unix System Services and nowadays z/OS UNIX.



IBM z/OS UNIX manuals

- z/OS 1.12 information center
  - SA22-7802-13 Command Reference
  - SA22-7807-11 Messages and Codes
  - GA22-7800-18 Planning
  - SA22-7805-08 Programming Tools
  - SA22-7801-13 User's Guide

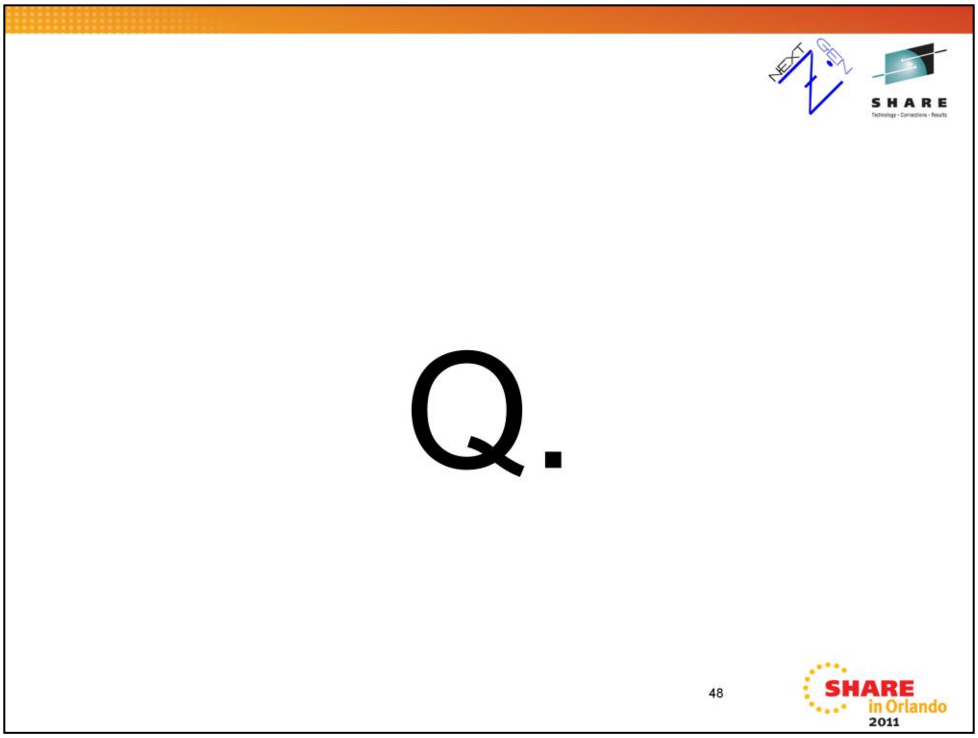
  - … and more at … http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.bpx/bpx.htm



Thank you for your attention!

Please do not forget to fill in evaluation forms.

Session 10067



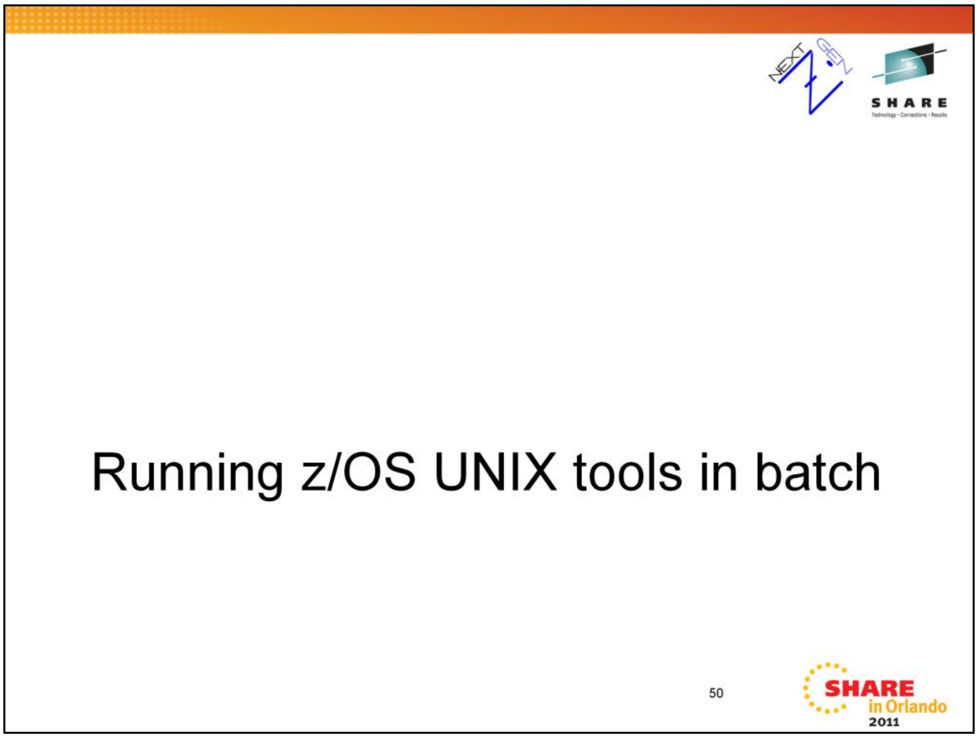
Q.

SHARE in Orlando 2011

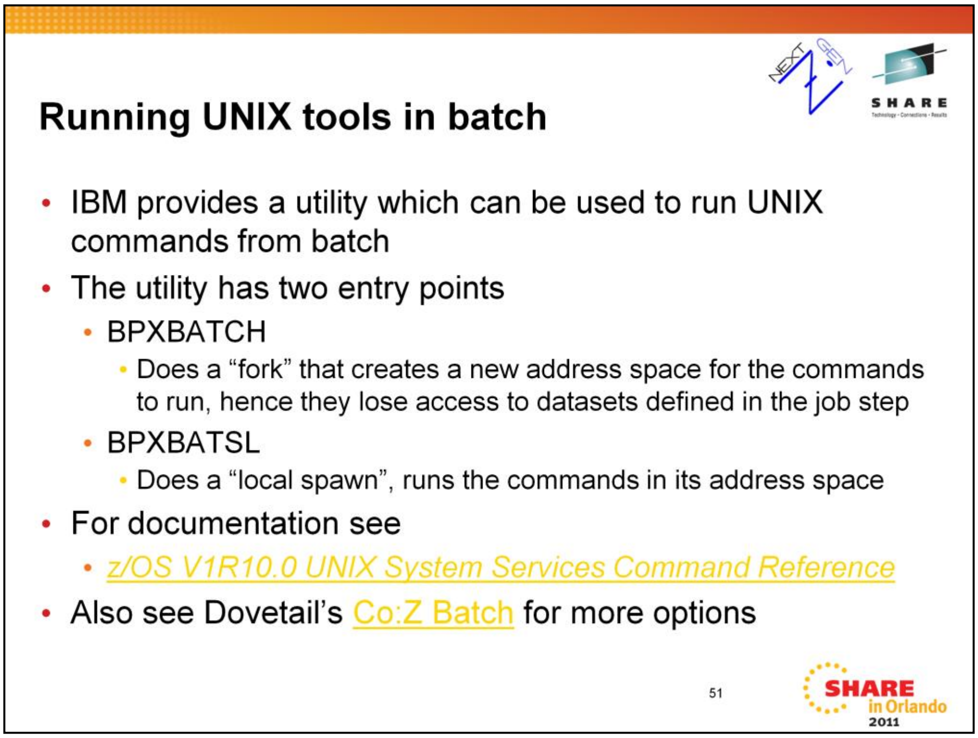
Additional material about running z/OS UNIX commands within batch is appended at the end of the session slides. It is included in proceedings.

The Next Generation of Mainframe Professionals.

Additional Material Follows



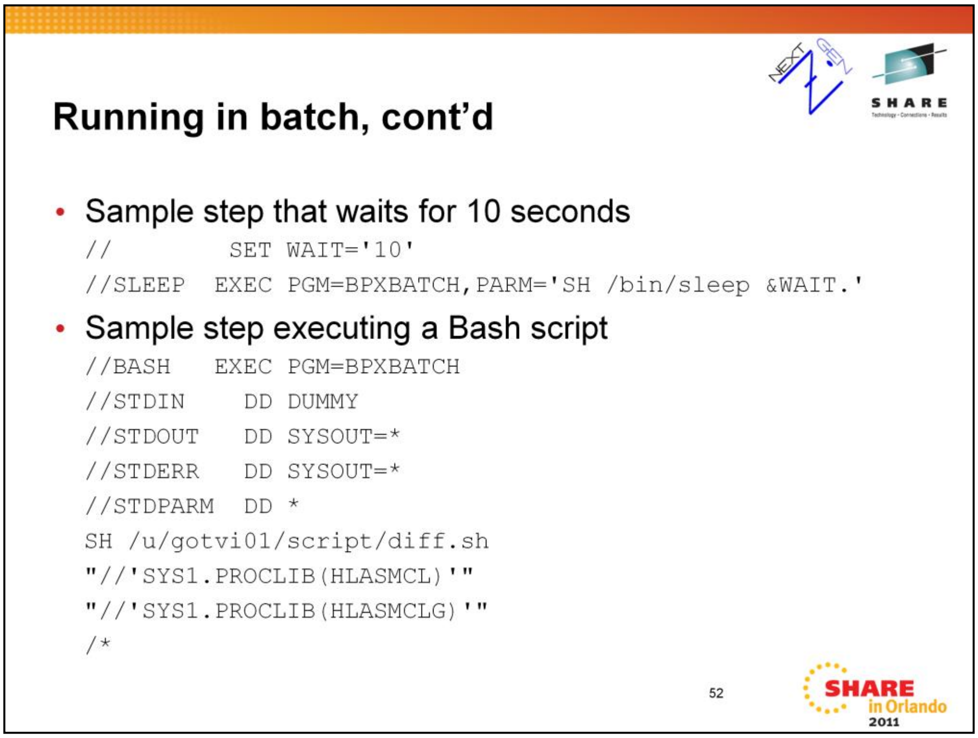
Running z/OS UNIX tools in batch



Running UNIX tools in batch

- IBM provides a utility which can be used to run UNIX commands from batch
- The utility has two entry points
  - BPXBATCH
    - Does a “fork” that creates a new address space for the commands to run, hence they lose access to datasets defined in the job step
  - BPXBATSL
    - Does a “local spawn”, runs the commands in its address space
- For documentation see
  - *z/OS V1R10.0 UNIX System Services Command Reference*
- Also see Dovetail’s Co:Z Batch for more options

See also www.dovetail.com an their COZBATCH utility, it’s free and has some advantages over BPXBATSL.



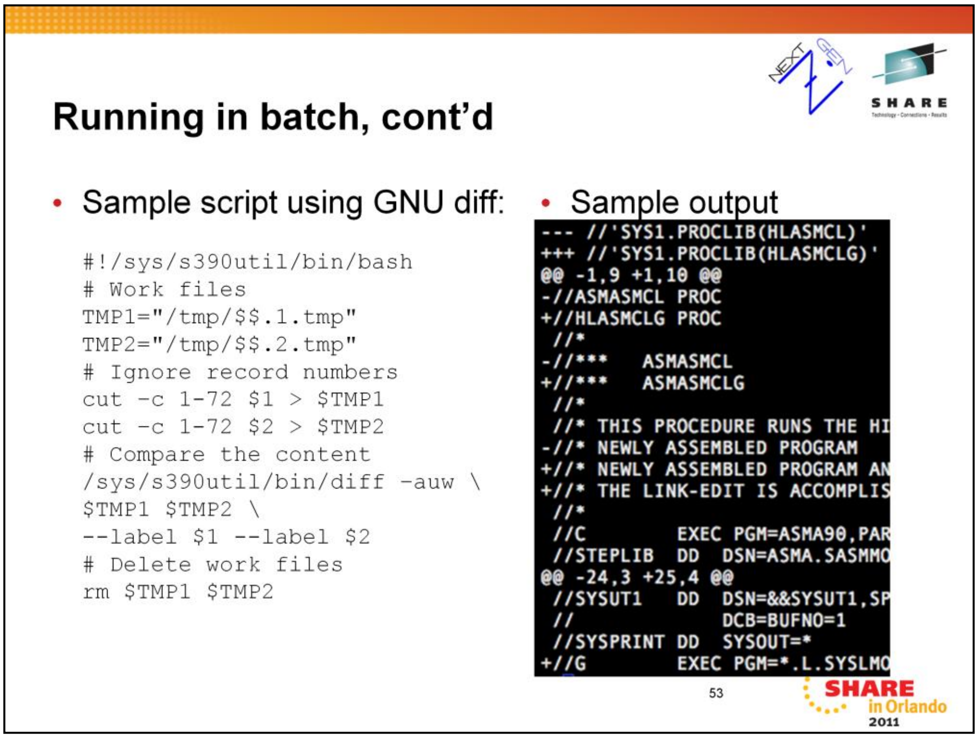
Running in batch, cont'd

- Sample step that waits for 10 seconds

```
//         SET WAIT='10'
//SLEEP  EXEC PGM=BPXBATCH,PARM='SH /bin/sleep &WAIT.'
```

- Sample step executing a Bash script

```
//BASH   EXEC PGM=BPXBATCH
//STDIN    DD DUMMY
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDPARM  DD *
SH /u/gotvi01/script/diff.sh
"//'SYS1.PROCLIB(HLASMCL)'"
"//'SYS1.PROCLIB(HLASMCLG)'"
/*
```



Running in batch, cont'd

- Sample script using GNU diff:

```
#!/sys/s390util/bin/bash
# Work files
TMP1="/tmp/$$.1.tmp"
TMP2="/tmp/$$.2.tmp"
# Ignore record numbers
cut -c 1-72 $1 > $TMP1
cut -c 1-72 $2 > $TMP2
# Compare the content
/sys/s390util/bin/diff -auw \
$TMP1 $TMP2 \
--label $1 --label $2
# Delete work files
rm $TMP1 $TMP2
```

- Sample output

```
--- //'SYS1.PROCLIB(HLASMCL)'
+++ //'SYS1.PROCLIB(HLASMCLG)'
@@ -1,9 +1,10 @@
-//ASMASMCL PROC
+//HLASMCLG PROC
 //*
-//***   ASMASMCL
+//***   ASMASMCLG
 //*
 //* THIS PROCEDURE RUNS THE HI
-//* NEWLY ASSEMBLED PROGRAM
+//* NEWLY ASSEMBLED PROGRAM AN
+//* THE LINK-EDIT IS ACCOMPLIS
 //*
 //C        EXEC PGM=ASMA90,PAR
 //STEPLIB  DD  DSN=ASMA.SASMMO
@@ -24,3 +25,4 @@
 //SYSUT1   DD  DSN=&&SYSUT1,SP
 //            DCB=BUFNO=1
 //SYSPRINT DD  SYSOUT=*
+//G        EXEC PGM=*.L.SYSLMO
```

Sample script `/u/gotvi01/script/diff.sh – compare content of two JCL procedures with GNU diff`

In most shells, $$ is a variable representing currently running process' PID. It helps us generate random, unique file names for temporary files.

$1 is the value of first parameter passed during script invocation. $2, $3, … $n are the respective n-the parameter. $0 is the command string used to invoke the script

Backslash '\' operates as a line continuation character.

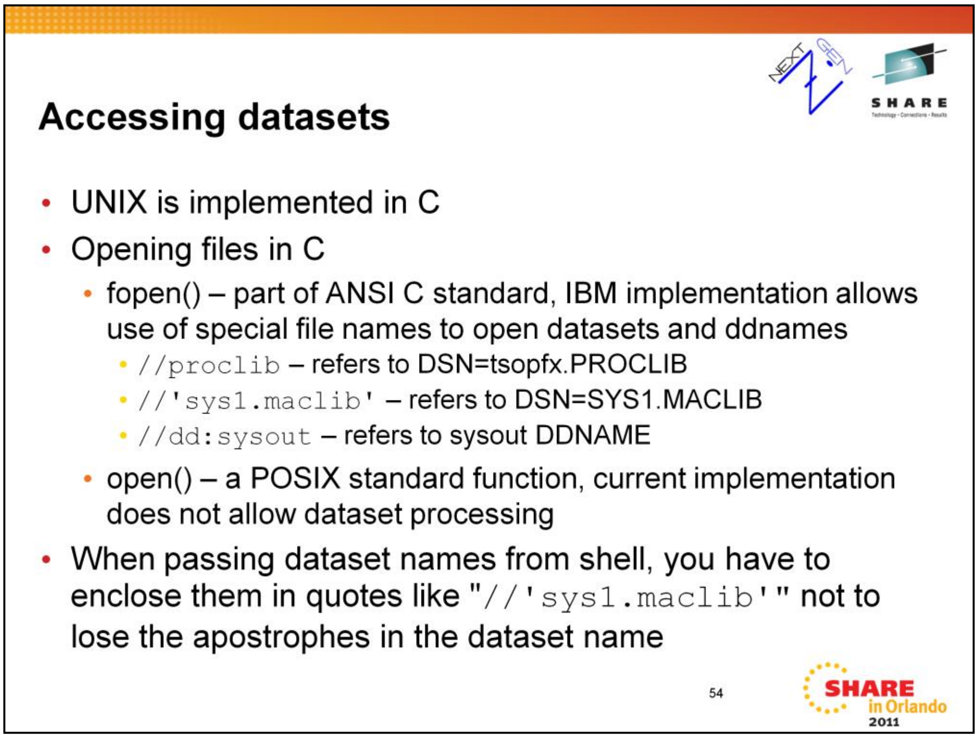
-auw tells GNU diff to

a) treat input as text files even when it encounters non-printable characters

u) use "universal diff" format for output (a de facto standard in open source community)

w) ignore all white spaces (space is equivalent to tab and so is any sequence of them)

beware that the port of GNU diff to z/OS UNIX suffers from some non-functional options (e.g. diff --help tells the user that --help is an unknown parameter and to type diff --help for help) however these are of low practical importance (help can be found on the GNU project documentation web page)



Accessing datasets

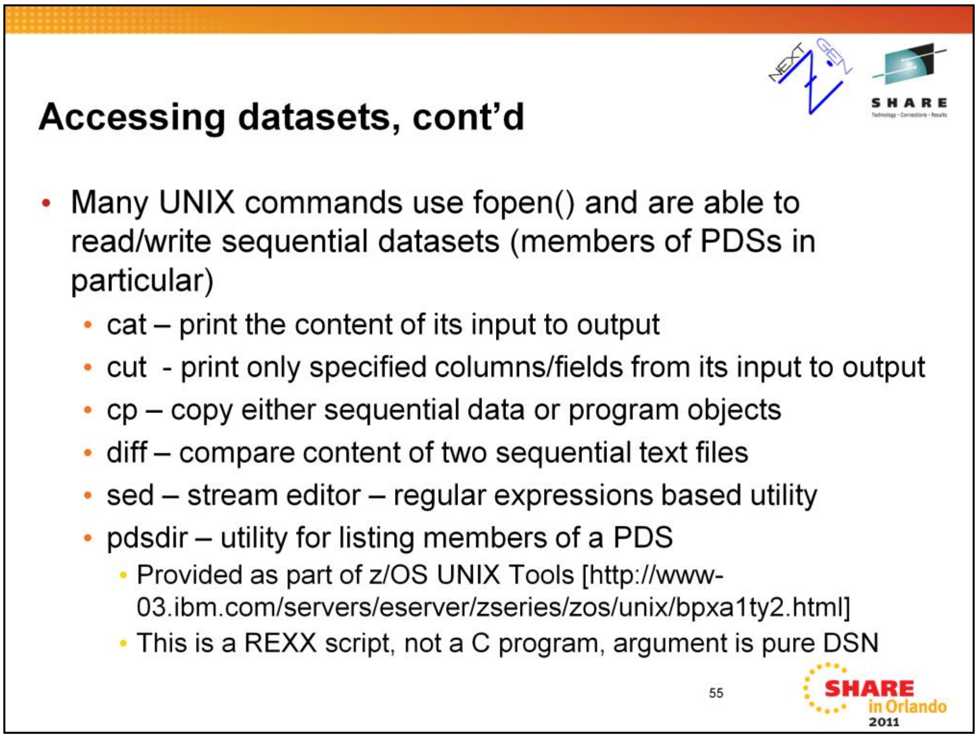
- UNIX is implemented in C
- Opening files in C
  - fopen() – part of ANSI C standard, IBM implementation allows use of special file names to open datasets and ddnames
    - `//proclib` – refers to DSN=tsopfx.PROCLIB
    - `//'sys1.maclib'` – refers to DSN=SYS1.MACLIB
    - `//dd:sysout` – refers to sysout DDNAME
  - open() – a POSIX standard function, current implementation does not allow dataset processing
- When passing dataset names from shell, you have to enclose them in quotes like `"//'sys1.maclib'"` not to lose the apostrophes in the dataset name

SHARE in Orlando 2011

For more details regarding IBM XL C/C++ runtime library see

z/OS V1R10.0 XL C/C++ Programming Guide (2.9.1.1 Using fopen() or freopen())

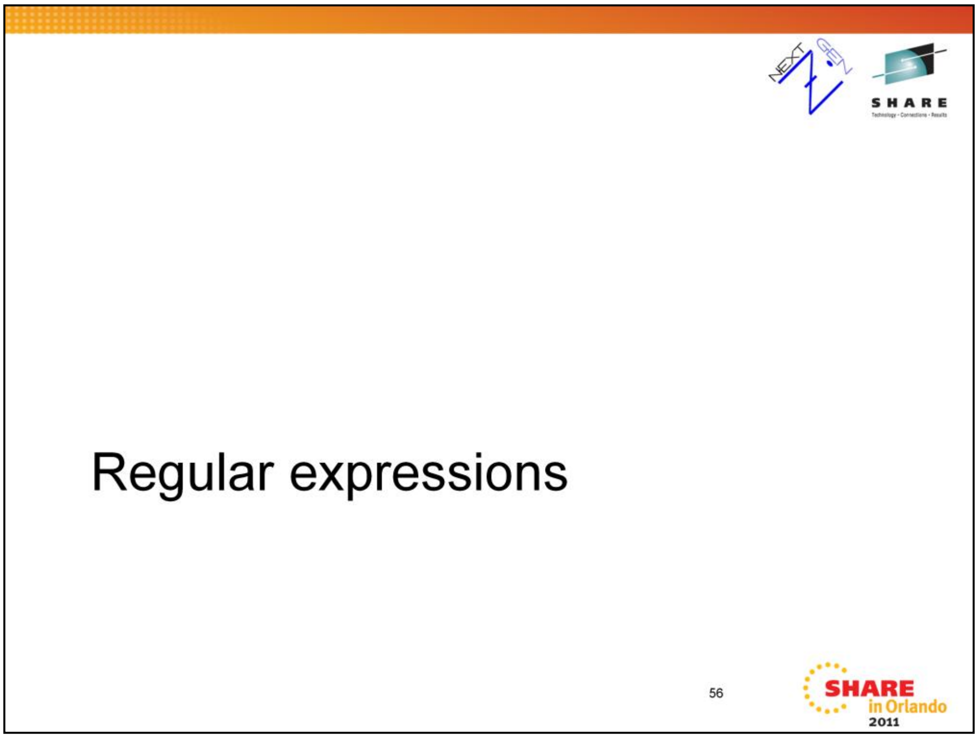
[http://publib.boulder.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.cbcpx01/ufopen.htm#wq269]



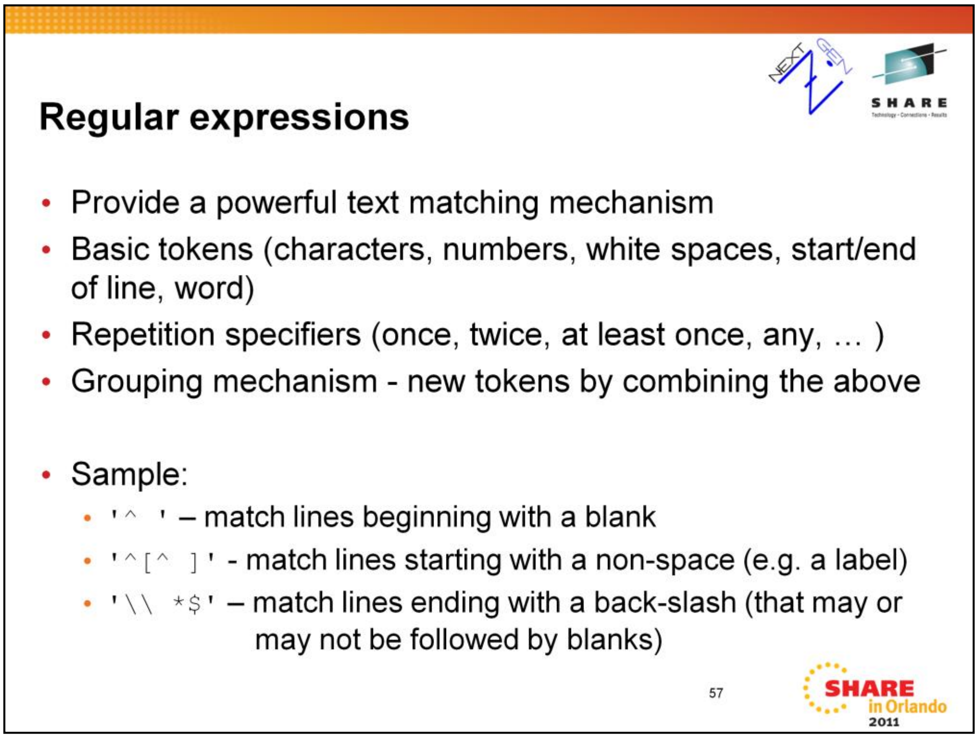
Accessing datasets, cont’d

- Many UNIX commands use fopen() and are able to read/write sequential datasets (members of PDSs in particular)
  - cat – print the content of its input to output
  - cut - print only specified columns/fields from its input to output
  - cp – copy either sequential data or program objects
  - diff – compare content of two sequential text files
  - sed – stream editor – regular expressions based utility
  - pdsdir – utility for listing members of a PDS
    - Provided as part of z/OS UNIX Tools [http://www-03.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty2.html]
    - This is a REXX script, not a C program, argument is pure DSN

The information on this slide is solely derived from the author’s experimenting with various utilities and may be wrong. It is certainly NOT complete and serves only as a hint for you what to try.



Regular expressions



Regular expressions

- Provide a powerful text matching mechanism
- Basic tokens (characters, numbers, white spaces, start/end of line, word)
- Repetition specifiers (once, twice, at least once, any, …)
- Grouping mechanism - new tokens by combining the above
- Sample:
  - `'^ '` – match lines beginning with a blank
  - `'^[^ ]'` - match lines starting with a non-space (e.g. a label)
  - `'\\ *$'` – match lines ending with a back-slash (that may or may not be followed by blanks)

57

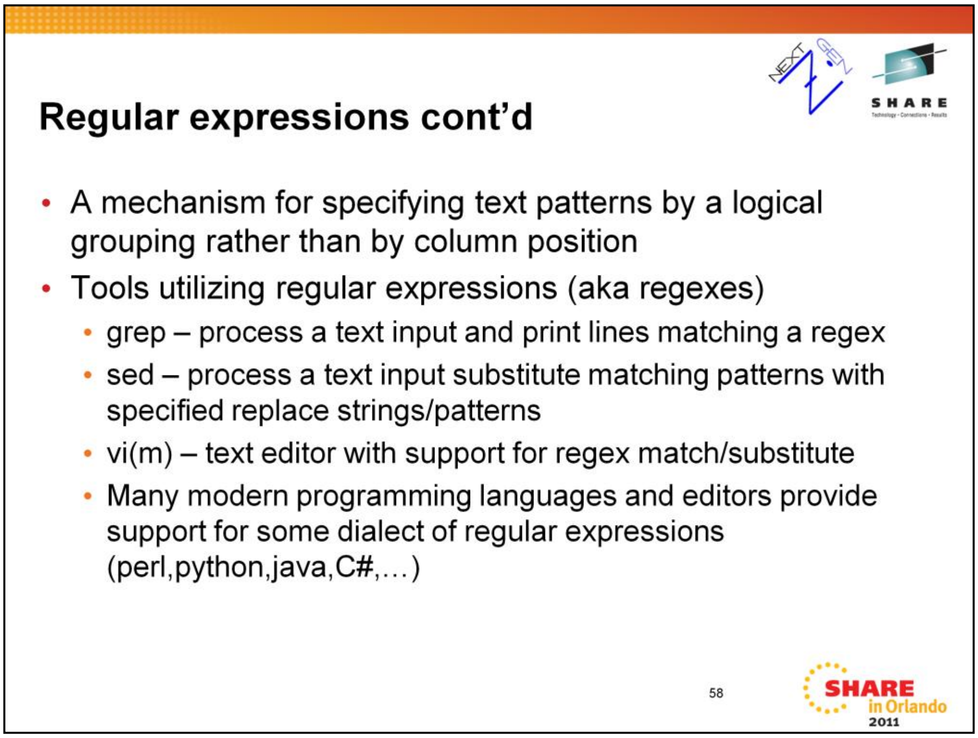
SHARE in Orlando 2011

The description here is overly simplified for the purpose of this presentation. There are many dialects of regular expressions.

For more information about IBM implementation see

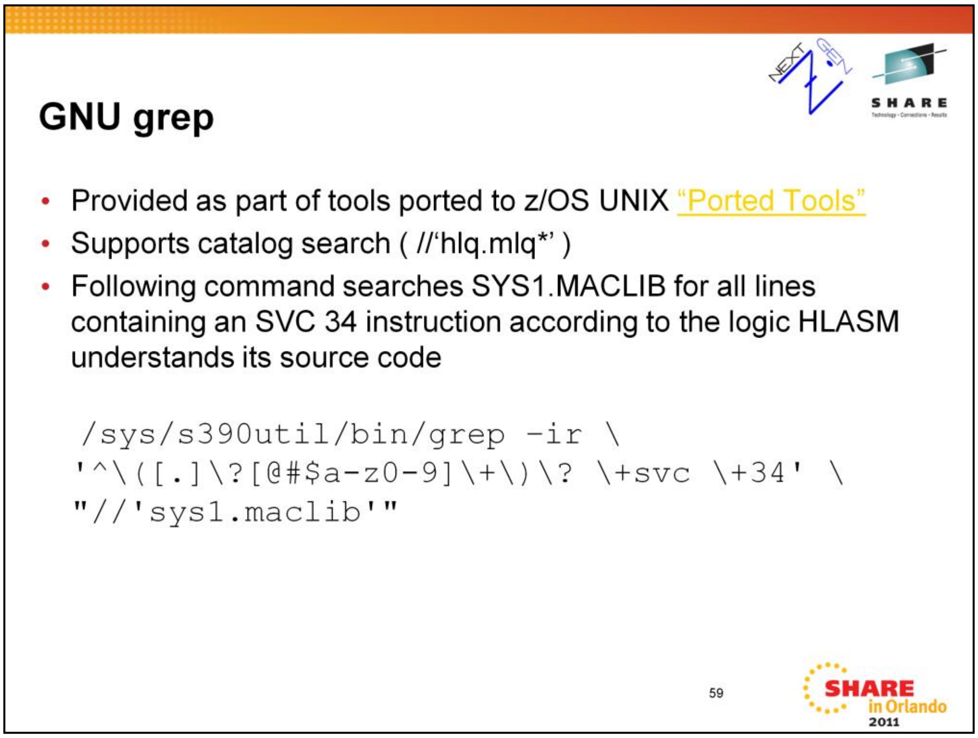
[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/BPXZA580/REGEXPA?SHELF=BPXZSH81&DT=20070503124658]

For more information about regular expressions and their various dialects see the book “Mastering regular expressions” by Jeffrey E.F. Friedl, O'Reilly Media1997



Regular expressions cont'd

- A mechanism for specifying text patterns by a logical grouping rather than by column position
- Tools utilizing regular expressions (aka regexes)
  - grep – process a text input and print lines matching a regex
  - sed – process a text input substitute matching patterns with specified replace strings/patterns
  - vi(m) – text editor with support for regex match/substitute
  - Many modern programming languages and editors provide support for some dialect of regular expressions (perl,python,java,C#,…)

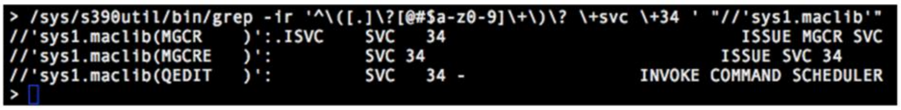


GNU grep

- Provided as part of tools ported to z/OS UNIX “Ported Tools”
- Supports catalog search (//‘hlq.mlq*’)
- Following command searches SYS1.MACLIB for all lines containing an SVC 34 instruction according to the logic HLASM understands its source code

```
 /sys/s390util/bin/grep -ir \
'^\([.]\?[@#$a-z0-9]\+\)\? \+svc \+34' \
"//'sys1.maclib'"
```

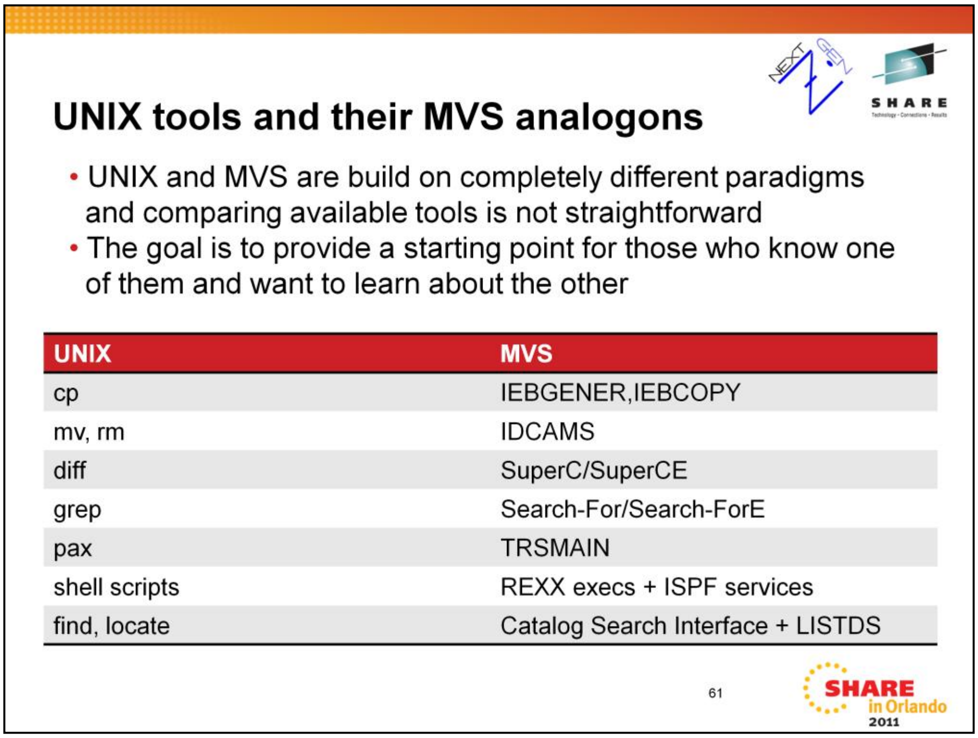
GNU grep cont'd



```
> /sys/s390util/bin/grep -ir '^\([.]\?[@#$a-z0-9]\+\)\? \+svc \+34 ' "//'sys1.maclib'"
//'sys1.maclib(MGCR    )':.ISVC    SVC   34                             ISSUE MGCR SVC
//'sys1.maclib(MGCRE   )':         SVC 34                             ISSUE SVC 34
//'sys1.maclib(QEDIT   )':         SVC   34 -                 INVOKE COMMAND SCHEDULER
> 
```

60

To display the line numbers where the particular match occurred use the -n option.



UNIX tools and their MVS analogons

- UNIX and MVS are build on completely different paradigms and comparing available tools is not straightforward
- The goal is to provide a starting point for those who know one of them and want to learn about the other

UNIX	MVS
cp	IEBGENER,IEBCOPY
mv, rm	IDCAMS
diff	SuperC/SuperCE
grep	Search-For/Search-ForE
pax	TRSMAIN
shell scripts	REXX execs + ISPF services
find, locate	Catalog Search Interface + LISTDS

Documentation for

IEBGENER, IEBCOPY, IEBUPDTE

z/OS V1R10.0 DFSMSdfp Utilities

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/DGT2U140/CCONTENTS?DT=20080521090625]

SuperC/SuperCE,Search-For/SearchForE

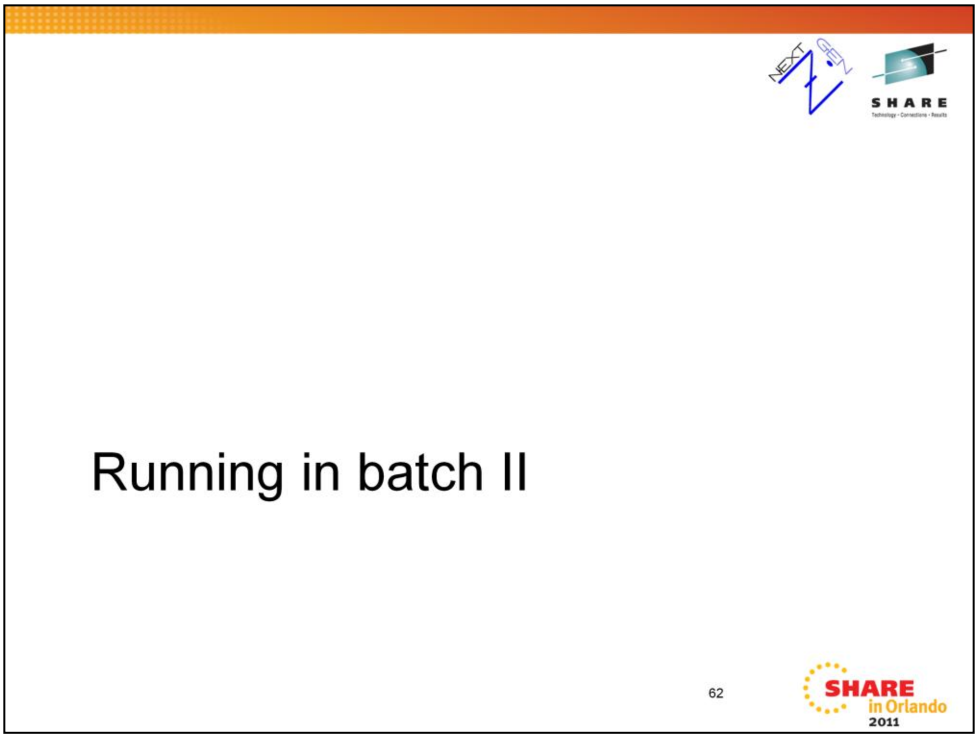
z/OS V1R10.0 ISPF User's Guide Vol II

[http://publib.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ISPZU270/CCONTENTS?SHELF=ISPZPM70&DN=SC34-4823-08&DT=20080613002317]

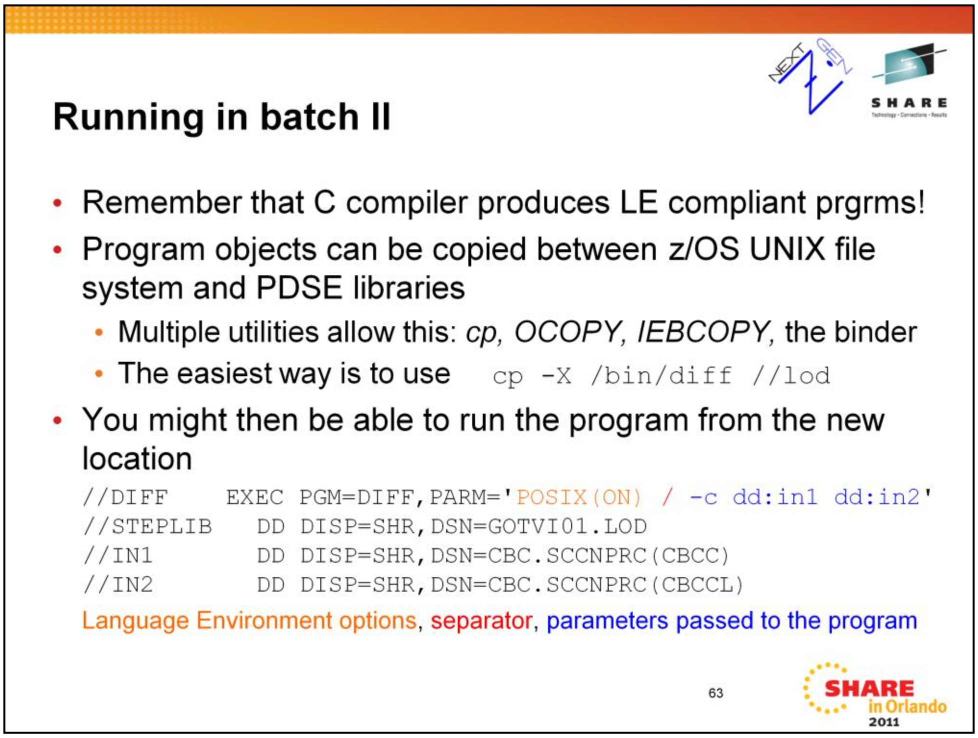
IDCAMS

z/OS V1R10.0 DFSMS AMS for Catalogs

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/DGT2I270/CCONTENTS?DT=20080602132758]



Running in batch II



Running in batch II

- Remember that C compiler produces LE compliant prgrms!
- Program objects can be copied between z/OS UNIX file system and PDSE libraries
  - Multiple utilities allow this: *cp, OCOPY, IEBCOPY,* the binder
  - The easiest way is to use `cp -X /bin/diff //lod`
- You might then be able to run the program from the new location

```
//DIFF     EXEC PGM=DIFF,PARM='POSIX(ON) / -c dd:in1 dd:in2'
//STEPLIB    DD DISP=SHR,DSN=GOTVI01.LOD
//IN1        DD DISP=SHR,DSN=CBC.SCCNPRC(CBCC)
//IN2        DD DISP=SHR,DSN=CBC.SCCNPRC(CBCCL)
```

Language Environment options, separator, parameters passed to the program

63

SHARE in Orlando 2011

For more information how to copy data and executables between z/OS UNIX file system and datasets see:

z/OS V1R10.0 UNIX System Services User's Guide, 21.0 Copying data between the z/OS UNIX file system and MVS data sets

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/BPXZA490/21.0?SHELF=BPXZSH91&DT=20080530153853]

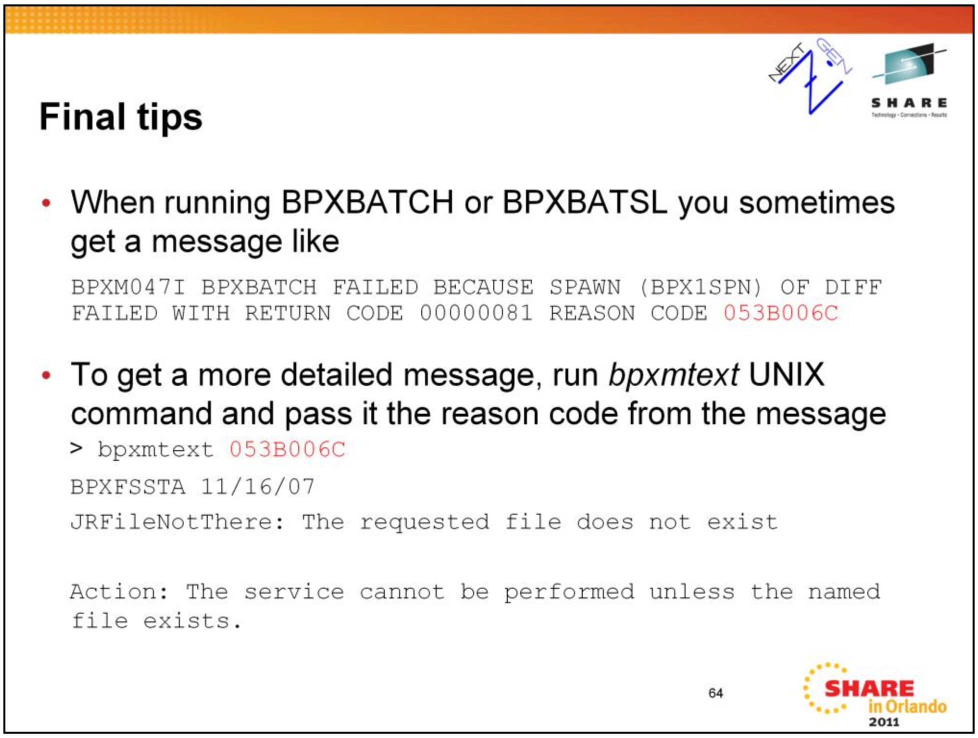
For more information about Language Environment and passing parameters to LE compliant programs see:

z/OS V1R1.0 Lang Env Prog Guide, 1.5.1.3 Specifying Run-Time Options in the EXEC Statement

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/CEEA2100/1.5.1.3?SHELF=CEE2BK00&DT=20010116123540]

IBM XL C/C++ compiler produces LE compliant code by default. You can change the default behavior through EXECOPS / NOEXECOPS compiler options or #pragma runopts macros.

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ceea2190/1.9.2?DT=20080603042013]



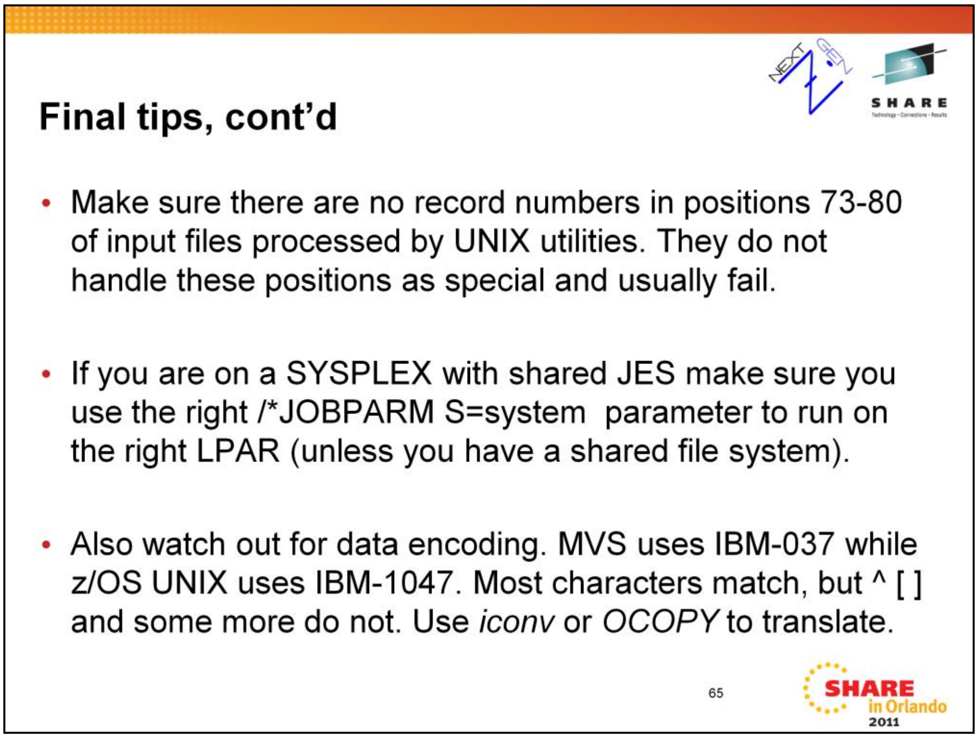
Final tips

- When running BPXBATCH or BPXBATSL you sometimes get a message like

```
BPXM047I BPXBATCH FAILED BECAUSE SPAWN (BPX1SPN) OF DIFF
FAILED WITH RETURN CODE 00000081 REASON CODE 053B006C
```

- To get a more detailed message, run *bpxmtext* UNIX command and pass it the reason code from the message

```
> bpxmtext 053B006C
BPXFSSTA 11/16/07
JRFileNotThere: The requested file does not exist

Action: The service cannot be performed unless the named
file exists.
```



Final tips, cont’d

- Make sure there are no record numbers in positions 73-80 of input files processed by UNIX utilities. They do not handle these positions as special and usually fail.

- If you are on a SYSPLEX with shared JES make sure you use the right /*JOBPARM S=system parameter to run on the right LPAR (unless you have a shared file system).

- Also watch out for data encoding. MVS uses IBM-037 while z/OS UNIX uses IBM-1047. Most characters match, but ^ [ ] and some more do not. Use *iconv* or *OCOPY* to translate.