

QShell You Can Use

Bruce Guetzkow

Developing on IBMi & predecessors
For 25+ years using
RPG, CL, SQL, REXX, QShell,
PHP, HTML, CSS, JavaScript,
& anything else I can get my hands on

QShell You Can Use

Please remember to
put mobile phones in
Stealth Mode

Thank you!

QShell You Can Use

- Agenda
 - What is QShell?
 - Commands You Can Use
 - Streams & Redirection
 - Pipelines
 - Accessing DB2 via QShell
 - Questions? Anytime!

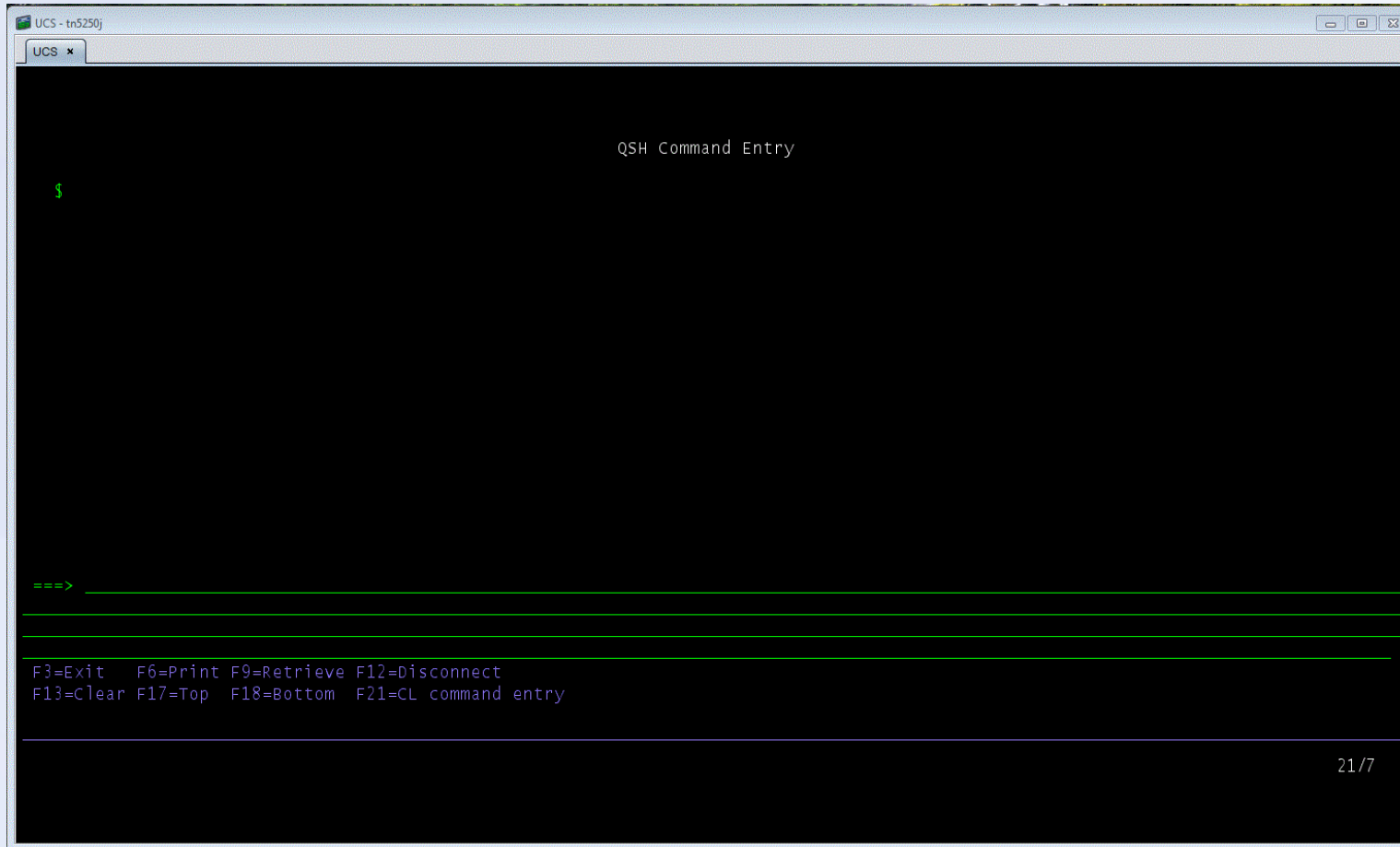
What is QShell?

- QShell is a command environment
- Terminal where Unix commands can be entered
- A way to interact with the IFS
- /QSYS.LIB is a part of the IFS

Getting Starting with QShell

- QSH or STRQSH
- One parameter: CMD
 - If CMD contains a value it is executed immediately
 - Otherwise, QShell Environment
 - Like FTP
 - Enter commands interactively

Getting Starting with QShell



The screenshot shows a terminal window titled "UCS - tn5250j" with a tab labeled "UCS x". The main content area is black with white text. At the top, it says "QSH Command Entry". Below that is a green prompt character "\$". Further down, there is a green "===>" prompt. At the bottom, a list of function key shortcuts is displayed: F3=Exit, F6=Print, F9=Retrieve, F12=Disconnect, F13=Clear, F17=Top, F18=Bottom, and F21=CL command entry. In the bottom right corner of the terminal, the text "21/7" is visible.

```
QSH Command Entry

$

===>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry

21/7
```

Getting Started with QShell

- Examples:
 - QSH CMD('ls /somefolder')
 - QSH CMD('ls /somefolder > /myfile.txt')
- Exit QShell
 - F3 – Closes terminal, ends session
 - F12 – Closes terminal, suspends session
 - Next QSH Resumes

Utility commands

- Commands to run against files in IFS
- Names are sometimes obtuse
 - Not like IBM commands!
- Format
 - Command
 - Options
 - Arguments

Utilities

- `ls` – List directory contents
- `grep` – Search a file for a pattern
- `wc` – Word, line and byte/character count
- `cut` – Cut out selected fields of each line of a file
- `sort` – Sort, merge or sequence check text files

Utilities

- `uniq` – Report or filter out repeated lines in a file
- `dirname` – Return directory portion of path name
- `basename` - Return non-directory portion of path name

Utilities

- head – Copy the first part of files
- tail – Display the last part of a file
- ajar – Alternative Java Archive
- system – Run CL command
- find – Find files

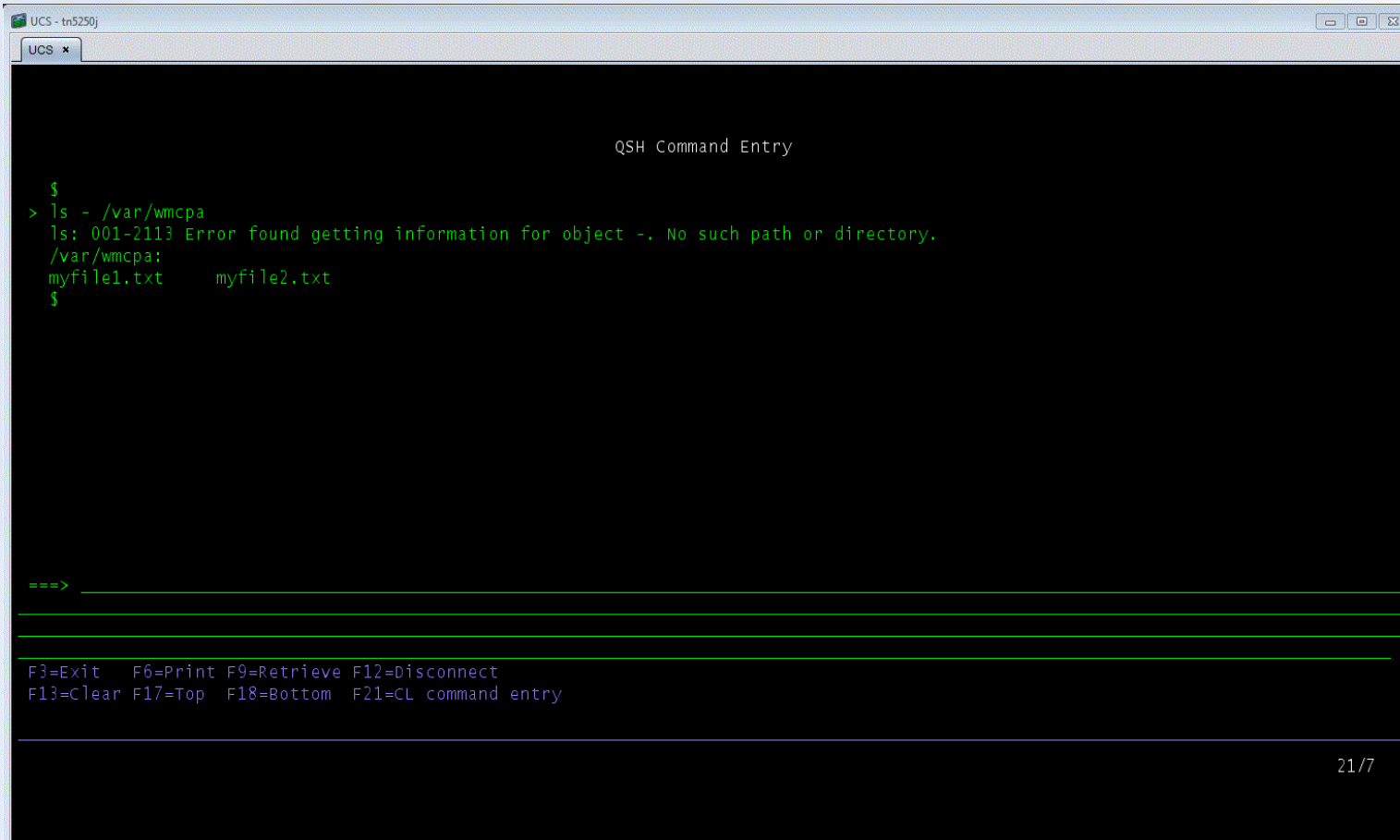
ls – List directory contents

- Use ls in QShell as you would use dir in FTP
- Useful options
 - -R: Recursively list sub-directories
 - -l: (lower-case L) Long format
 - -t: Sort by time modified (most recent first)

ls – List directory contents

- `ls -l`
 - List current directory in Long Format
- `ls -t /path/to/directory`
 - List indicated directory by time modified
- `ls -lt /path/to/directory/abc*.*`
 - List all files beginning with name using wildcards in indicated directory in Long Format by time modified

ls – List directory contents



The screenshot shows a QShell terminal window titled "UCS - tn5250j" with a tab labeled "UCS *". The terminal displays the following text:

```
QSH Command Entry

$
> ls - /var/wmcpa
ls: 001-2113 Error found getting information for object -. No such path or directory.
/var/wmcpa:
myfile1.txt    myfile2.txt
$

===>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top   F18=Bottom  F21=CL command entry

21/7
```

grep – Search a file for a pattern

- Search contents of files for text string
- Useful options
 - -R: Recursively search sub-directories
 - -c: Count selected lines
 - -e: Search for list of patterns
 - -l: (lower-case L) List file name(s) only

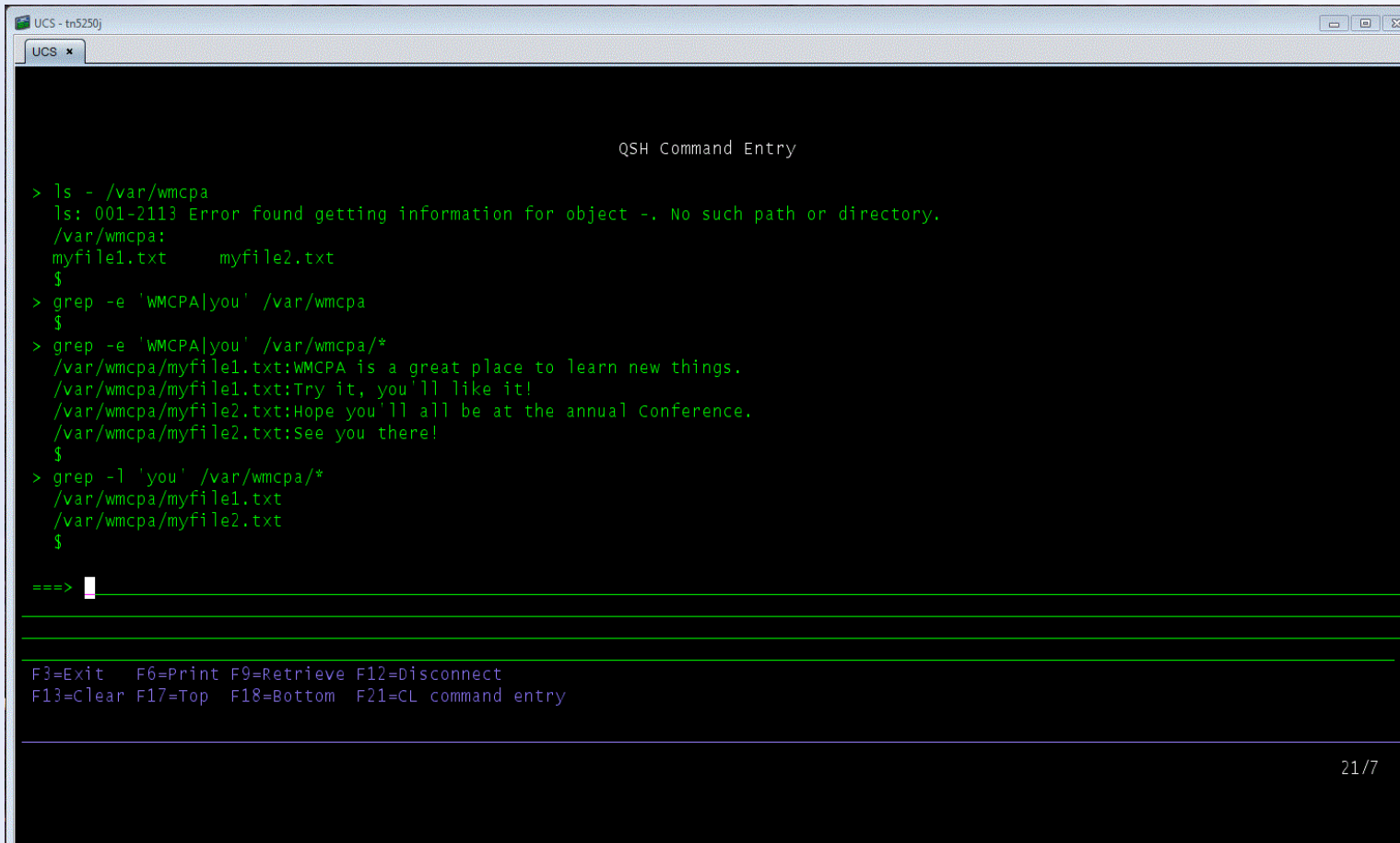
grep – Search a file for a pattern

- `grep -R 'text string' /path/to/directory`
 - List all lines in all files in specified directory and sub-directories containing text string
- `grep -l 'text string' /path/to/directory`
 - List file names in specified directory containing text string

grep – Search a file for a pattern

- `grep -c 'text string' /path/to/directory/*`
 - Return number of lines in all files in indicated directory containing text string
- `grep -l -e 'string1|string2|string3' /path/to/directory/*`
 - List file names in indicated directory containing any of the strings listed

grep – Search a file for a pattern



```
UCS - tn5250j
UCS *
QSH Command Entry

> ls - /var/wmcpa
ls: 001-2113 Error found getting information for object -. No such path or directory.
/var/wmcpa:
myfile1.txt    myfile2.txt
$
> grep -e 'wMCPA|you' /var/wmcpa
$
> grep -e 'wMCPA|you' /var/wmcpa/*
/var/wmcpa/myfile1.txt:wMCPA is a great place to learn new things.
/var/wmcpa/myfile1.txt:Try it, you'll like it!
/var/wmcpa/myfile2.txt:Hope you'll all be at the annual Conference.
/var/wmcpa/myfile2.txt:See you there!
$
> grep -l 'you' /var/wmcpa/*
/var/wmcpa/myfile1.txt
/var/wmcpa/myfile2.txt
$

===> |

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top   F18=Bottom  F21=CL command entry

21/7
```

wc – Word, line and byte/character count

- List the number of words, lines, bytes and characters in a file
 - Word: characters separated by white space
 - Line: characters separated by new line
- `wc -w /path/to/file.txt`
 - List the number of words in file indicated

cut – Cut out selected fields of each line of a file

- Extract specified segments from indicated file
- Useful Options
 - -b: Specify list of byte positions

cut – Cut out selected fields of each line of a file

- `cut -b '21-30' /path/to/file.txt`
 - Extract positions 21-30 of indicated file
- `cut -b '21-30,171-191,45-59' /path/to/file.txt`
 - Extract positions 21-30, 171-191 and 45-59
 - Data is concatenated together

sort – Sort, merge or sequence check text files

- Sort contents of file by line
- Useful Options
 - -k: Key definitions
 - -o: Output file
 - None: Sort based on entire line
- Field = characters terminated by white space or new line character

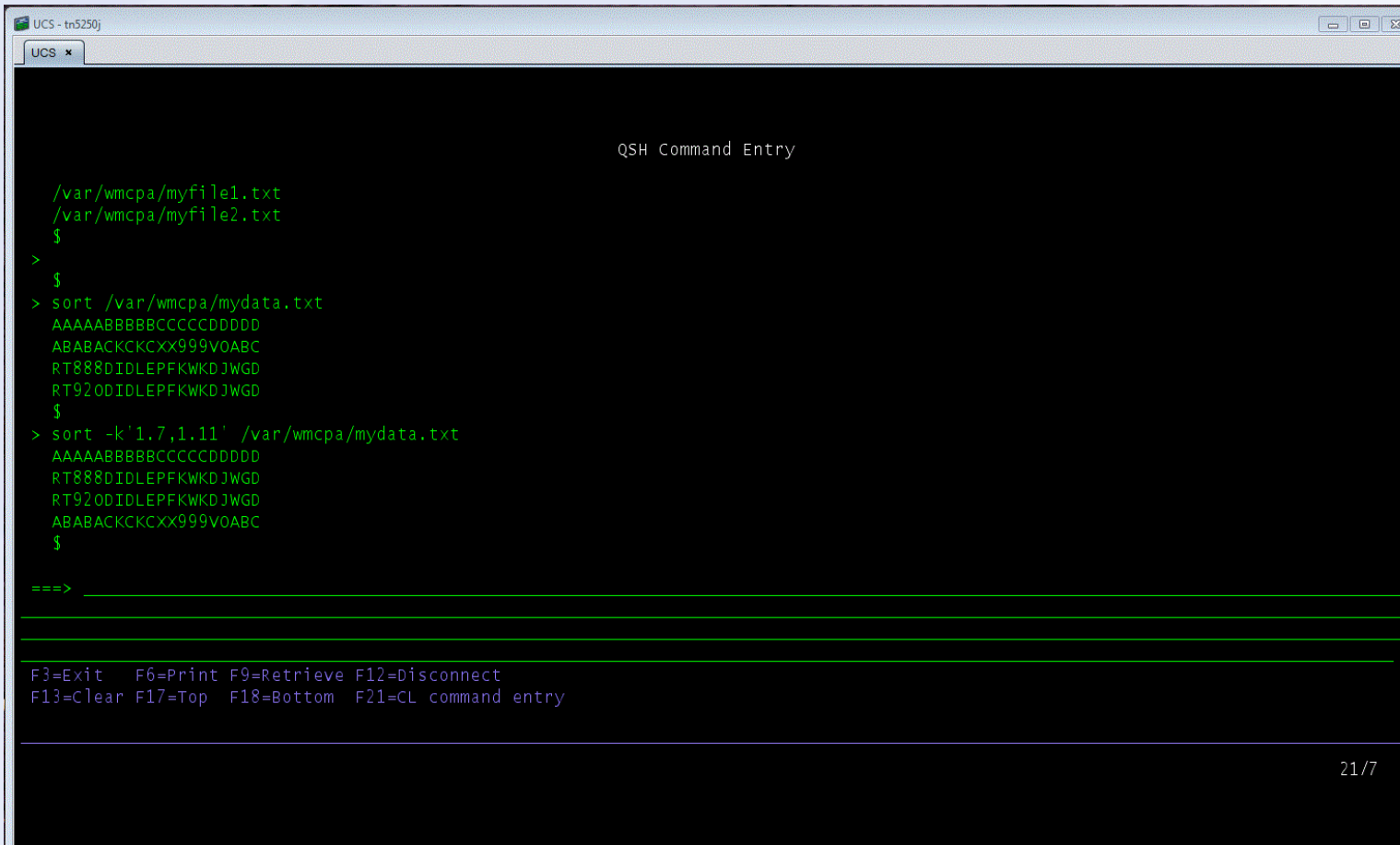
sort – Sort, merge or sequence check text files

- `sort /path/to/data.txt`
 - Sort based on entire line
- `sort -k'1.21,1.30' /path/to/data.txt`
 - Sort on key field 1 from positions 21-30
- `sort -k'1.21,1.30' -k'1.45,1.55' /path/to/data.txt`
 - Sort on 2-part key: 21-30 and 45-55

sort – Sort, merge or sequence check text files

- `sort -k'1.10,1.15' /path/to/data.txt -o /path/to/results.txt`
 - Sort by key field 1 positions 10-15 and write to file indicated after “o” option

sort – Sort, merge or sequence check text files



```
UCS - tn5250j
UCS *
QSH Command Entry

/var/wmcpa/myfile1.txt
/var/wmcpa/myfile2.txt
$
>
$
> sort /var/wmcpa/mydata.txt
AAAAABBBBCCCCDDDD
ABABACKKCXX999V0ABC
RT888DIDLEPFKWDJWGD
RT920DIDLEPFKWDJWGD
$
> sort -k'1.7,1.11' /var/wmcpa/mydata.txt
AAAAABBBBCCCCDDDD
RT888DIDLEPFKWDJWGD
RT920DIDLEPFKWDJWGD
ABABACKKCXX999V0ABC
$

===>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry

21/7
```

uniq – Report or filter out repeated lines in a file

- Read file, write unique lines
- Useful Options
 - -c: Precede output line with count
 - -d: Suppress non-duplicated lines
 - -s: Ignore first “number of characters”
 - -u: Suppress duplicated lines

uniq – Report or filter out repeated lines in a file

- `uniq -s 20 /path/to/input.txt /path/to/output.txt`
 - input.txt is read
 - A line is a duplicate if positions 21-end are identical
 - Unique copies of lines are written to output.txt

dirname – Return directory portion of path name

- Retrieve the directory portion of a path name
- No Options
- `dirname /path/to/file.txt`
 - `/path/to`

basename - Return non-directory portion of path name

- Retrieve the non-directory portion of a path name
- No Options
- `basename /path/to/file.txt`
 - `file.txt`

head – Copy the first part of files

- Display first part of file
- Useful Options
 - -n: Indicate number of lines
 - Default is 10 lines
- `head -n 20 /path/to/file.txt`
 - List first 20 lines of file

tail – Display the last part of a file

- Display last part of file
- Useful Options
 - -n: Indicate number of lines
 - Default is 10 lines
- `tail -n 20 /path/to/file.txt`
 - List last 20 lines of file

ajar – Alternative Java Archive

- Create and Extract from Archive Files
- .jar, .zip, others
- Useful Options
 - -c: Create new archive file
 - -a: Add file to archive file
 - -x: Extract from archive file
 - -l: List contents of archive file

ajar – Alternative Java Archive

- `ajar -c myjar /path/to/folder`
 - Create jar file “myjar” containing files from /path/to/folder
- `ajar -c -r myzip.zip /path/to/folder`
 - Create zip file “myjar” containing files from /path/to/folder and all subdirectories
- `ajar -x myjar`
 - Extract all from myjar to current directory

system – Run CL command

- Run any CL command
- Useful Options
 - -k: Keep all spooled files (normally written to STDOUT and deleted)
 - -K: Keep all spooled files and joblog (normally written to STDOUT and deleted)
 - -q: Do not write messages to STDERR

system – Run CL command

- system – WRKACTJOB
- system – “CRTDIR DIR('/path/to/folder)’”
- system – “DLTF FILE(QGPL/SOMEFILE)’”

find – Find files

- Find list of files
- Useful Options
 - -P: No symbolic links are followed
- Primaries – Actions to take on files
 - -exec: Execute utility command
 - -ls: List information on Files
 - -name: Select files with name pattern

find – Find files

- `find /path/to/folder -name '*.txt'`
 - Find all files in directory /path/to/folder that have a file extension of .txt
- `find /path/to/folder -name '*.txt' -exec rm {} \;`
 - Find all files in directory /path/to/folder that have a file extension of .txt and delete them

find – Find files

- `find /path/to/folder`
`-exec grep -l 'text string' {} \;`
 - Find all files starting at directory `/path/to/folder`, list those file names containing 'text string'

Streams

- HLL Programs read/write from/to many objects:
 - Files, Data Areas, etc.
- QShell uses streams of data:
 - STDIN
 - STDOUT
 - STDERR

Streams

- Interactive default is terminal (screen) for all streams
- Enter command in terminal, QShell processes it
- Output is returned to terminal
- Errors also returned to terminal

Streams

- Descriptors
 - 0 – 9
 - 0 = STDIN
 - 1 = STDOUT
 - 2 = STDERR
 - Unless you define them, only these three are used

Streams

- Interactive
 - Input from Keyboard in Terminal
 - Output and Errors returned to Terminal
- Non-Interactive
 - Input from file STDIN
 - Output to QSYSPRT

Redirection

- Change where inputs/outputs come from and go to
 - < = Redirect STDIN
 - < /path/to/input.txt
 - > and >> = Redirect STDOUT
 - > /path/to/output.txt (overwrite)
 - >> /path/to/output.txt (append)

Redirection

- `ls -l > /path/to/output.txt`
 - List the contents of the indicated directory and write them to the file specified
- `ls -l >> /path/to/output.txt`
 - List the contents of the indicated directory and append them to the file specified

Redirection

- Redirect STDOUT and STDERR to same file
 - `ls -l > /path/to/output.txt 2>&1`
 - Can use Descriptor Number
 - 2 (STDERR) is redirected to 1 (STDOUT)

Pipelines

- Use the output of one command as the input to another
- Use Pipe: |

Pipelines

- Examples of using Pipelines
- `grep -R -f /path/to/selection_file.txt /path/to/directory | cut -b "1-35,41-47" | sort -r -k "1.36,1.42" -k "1.1,1.35" | uniq -s 35 | sort;`
- Results of “grep” are passed to “cut”, those results are passed to “sort”, then passed to “uniq”, finally to another “sort”

Accessing DB2 via QShell

- Simplest way is in a CL Program
 - OVRDBF FILE(STDIN)
TOFILE(INPUTFILE)
 - OVRDBF FILE(STDOUT)
TOFILE(OUTPUTFILE)
 - DLTOVR FILE(*ALL)

Accessing DB2 via QShell

- Use IFS-Name with Redirection
- `ls -l > /QSYS.LIB/MYLIB.LIB/MYFILE.FILE/MYMBR.MBR`
- Be careful when accessing DB2 data
 - QShell knows nothing about externally described files
 - Everything is a flat-file
 - You will get what you ask for!

Accessing DB2 via QShell

- `db2` – Run SQL statement
- `db2 select column from library.table`
- `db2 update library.table set column = value`

- `System` – Run CL Command
- `system "CRTLIB LIB(MYLIB)"`

Questions?

- QShell Manual
- Ted Holt Book

QShell You Can Use

Thank You!

Contact me at:

Twitter: @gmantechi

Email: gmantechi@gmantechconsulting.com