



# z/VM Tools

Mike MacIsaac, Sine Nomine Associates

[mmacisaac@sinenomine.net](mailto:mmacisaac@sinenomine.net)

VM Workshop, Columbus OH

June 14<sup>th</sup>, 2025



# Abstract

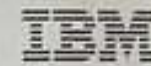
This session will discuss and demonstrate more than 20 REXX EXECs, many inspired by Linux commands. These include GREP, DIFF, HEAD, MAN, TAIL, WHICH, WC and WHO. They are open source and available on Github.

In addition a Web front end named z/VM and Linux Modern Administration (zlma) will be discussed and demonstrated.



# Bio

- Past jobs
  - IBM (27 yrs)
    - June 1984 (intern)
    - Dec 1997
    - Jan 2000
    - June 2005
    - Aug 2007
  - Innovation (2 yrs)
  - ADP (8 yrs)
- Current job: Sine Nomine Associates
  - Consulting for a financial institution



# z/VM and Linux on zSeries

From LPAR to Virtual Servers in Two Days

Follow steps based on z/VM  
virtualization technology

Set up your environment in two  
days; clone it in two minutes

Use 31-bit or 64-bit distributions,  
both addressed in this book



Michael MacIsaac  
Curtis Gearhart  
Jin Xiong

[ibm.com/redbooks](http://ibm.com/redbooks)

**Redbooks**



# Overview ...

- **The zvm-tools repository**
  - **Collection of scripts for z/VM sysadmins**
  - **Many inspired by Linux commands**
    - **diff**
    - **grep**
    - **head, tail**
    - **history**
    - **man**
    - **wc**
    - **which**
    - **who**



# ... Overview

- **Audience**
  - zLinux personnel learning z/VM
  - z/VM Sysadmins
- **Source:** [https://github.com/openmainframeproject/zvm-community-to](https://github.com/openmainframeproject/zvm-community-tools)
- **Documentation:** README with each EXEC described
- **License:** Apache 2.0
- **Contributions**
  - Please do!
  - Contact: [mike99mac@gmail.com](mailto:mike99mac@gmail.com)
- **How to use**
  - Copy ZVMTOOLS.VMARC to z/VM in binary
  - Unpack it on a tools disk
- **Summary**
  - Suite of EXECs for z/VM admins with documentation to streamline automate common z/VM tasks borrowing commands from Linux



# CALC EXEC

- Calculate a numerical expression
- Help:

```
calc -h
```

```
Name:  CALC - calculate a numerical expression
```

```
Usage: CALC expression
```

```
Exmp1: CALC 5 + 6
```

- The code

```
parse upper arg args
interpret "result =" args
say args "=" result
```



# CALCDASD EXEC

- **Compute CP owned and system attached DASD**
- **Help:**

`calcdasd -h`

**Name: CALCDASD EXEC - compute total disk space**

**Usage: CALCDASD**



# CALCOSA EXEC

- Report on all used and free OSA devices, including CHPIDs and PCHIDs
- Help:

```
calcosa -h
```

```
Name:  CALCOSA EXEC - compute OSA statistics
```

```
Usage: CALCOSA {-h|?} {(vd)}
```

```
Where: -h or '?' = give help
```

```
      v = verbose (include FREE OSAs)
```

```
      d = debug
```



# CFM EXEC

- **Copy a file, changing the file mode, but not file name nor type**
- **Help:**

```
cfm -h
```

```
Name: CFM EXEC - Copy file changing only file mode
```

```
Usage: CFM fn1 ft1 fm1 fm2 ['('options')']
```

```
Where: 'fn1 ft1 fm1' is the source file:
```

```
'fm2' is the target file mode
```

```
'options' add to COPY command such as 'REP' or 'OLDD'
```



# CFN EXEC

- **Copy a file, changing the file name, but not file type and mode**
- **Help:**

```
cfm -h
```

```
Name: CFN EXEC - Copy file changing only file name
```

```
Usage: CFN fn2 fn1 ft1 fm1 ['('options')']
```

```
Where: 'fn2' is the target file name
```

```
'fn1 ft1 fm1' is the source file:
```

```
'options' add to COPY command such as 'REP' or 'OLDD'
```



# CFT EXEC

- **Copy a file, changing the file type, but not file name and mode**
- **Help:**

```
cft -h
```

```
Name: CFT EXEC - Copy file changing only file type
```

```
Usage: CFT ft2 fn1 ft1 fm1 ['('options')']
```

```
Where: 'ft2' is the target file type
```

```
'fn1 ft1 fm1' is the source file:
```

```
'options' add to COPY command such as 'REP' or 'OLDD'
```



# COPYDISK EXEC

- Copy a minidisk - first try FLASHCOPY, if that fails use DDR
- Help:

`copydisk -h`

Name: COPYDISK EXEC - copy minidisk with FLASHCOPY or DDR

Usage: COPYDISK source\_vdev target\_vdev



# CPFORMAT EXEC

- Format many disks using CPFMTXA using a specific labeling convention
- Help:

Name: CPFORMAT EXEC

Format and label DASD as page, perm, spool or temp disk space

The label written to each DASD is J<t><xxxx> where:

<t> is type - P (page), M (perm), S (spool) or T (Temp disk)

<xxxx> is the 4 digit address

Syntax:

```
<-----<
>>--CPFORMAT--.-vdev-----.--AS---.-PERM-.------><
      '-vdev1-vdev2-'           '-PAGE-'
                                   '-SPOL-'
                                   '-TEMP-'
```



# DIFF EXEC

- Compare files using same algorithm and format as Linux
- Help:

`diff -h`

Name : DIFF EXEC - Compare two files

Usage: `diff fn1 ft1 fm1 fn2 ft2 fm2 (flags)`

Where: 'fn1 ft1 fm1' is the first file

'fn2 ft2 fm2' is the second file

flags can be: 'S': silent

'T': trace

'V': verbose



# GREP EXEC

Name: GREP EXEC - search files for text patterns

Usage: GREP [lrx-flags] 'pattern' FN FT [FM] [(flags)]

Where: pattern is a single quote delimited search string

FN FT is the file name and type to search

FM is the file mode (default A)

Where: zvm '(FLAGS' or Linux '-flags' can be one or more of:

i: ignore case

n: show line numbers

v: inverse - show non-matches

Special characters:

. - Matches one character

.\* - Matches one or more characters

^ - Matches start of line

\$ - Matches end of line

VM Workshop



# HEAD EXEC

- **Show the first part of one or more files**
- **Help**

**head -h**

**Name: HEAD EXEC - output the first part of files**

**Usage: HEAD [-n] fn ft [fm]**

**Where: 'fn ft' is the file name and type**

**: 'fm' is the file mode (default A)**



# HISTORY EXEC

- Show the history of commands issued previously w/logon/logoff times
- Help

`history -h`

Name: HISTORY EXEC - Show the command history

Usage: HISTORY [-lines] [pattern]

Where: '-lines' is optional number of tail lines to show

'pattern' is optional search pattern



# MAN EXEC

- Give z/VM help including subcommands QUERY and SET
- Help

man -h

Name: MAN EXEC - give help on command, details on QUERY and SET

Usage: MAN command

| MAN Query subcmd

| MAN SET subcmd

Where: command can be CMS, CP, XEDIT, TCPIP or REXX

subcmd can be CMS, CP or XEDIT Query or SET subcommands



# RFN EXEC

- **Rename a file, changing the file name, but not file type nor mode**
- **Help**

**rfn -h**

**Name: RFN EXEC - Rename file changing only file name**

**Usage: RFN fn2 fn1 ft1 fm1**

**Where: 'fn2' is the new file name**

**'fn1 ft1 fm1' is the source file**



# RFT EXEC

- **Rename a file, changing the file type, but not file name nor mode**
- **Help**

**rft -h**

**Name: RFT EXEC - Rename file changing only file type**

**Usage: RFT ft2 fn1 ft1 fm1**

**Where: 'ft2' is the new file type**

**: 'fn1 ft1 fm1' is the source file**



# RM EXEC

- **Remove file(s), supporting wild cards**
- **Help**

**Name: RM EXEC - erase one or more files**

**Usage: rm fn [ft [fm]]**

**Where: fn, ft or fm can be '\*' for all files**



# SSICMD EXEC

- Run CP commands on all LPARs
- Help:

```
ssicmd -h
```

Name: SSICMD EXEC - Issue a CP command on all SSI members

Usage: SSICMD <CPcmd>



# TAIL EXEC

- Show the ending lines of files
- Help:

```
tail -h
```

```
Name: TAIL EXEC - output the last part of file(s)
```

```
Usage: TAIL [-n] fn ft [fm] ...
```

```
Where: 'fn ft' is the file name and type
```

```
'fm' is the file mode (default A)
```



# WC EXEC

- **Count the number of lines, words and bytes in one or more files**
- **Help:**

`wc -h`

**Name :** WC EXEC - Print line, word & byte counts for each file

**Usage:** WC [OPTION] [FILE] ...

**Where:** OPTION can be: `-l`: show number of lines

`-w`: show number of words

`-c`: show number of characters



# WHICH EXEC

- Count the number of lines, words and bytes in one or more files
- Help:

Name: WHICH EXEC - list type of command, details on QUERY

Usage: WHICH command [(FLAGS)]

| WHICH Query subcommand ['(FLAGS)']

| WHICH SET subcommand ['(FLAGS)']

Where: FLAGS can be:

D: debug

H: create help command

S: silent - negative verbosity

T: trace

V: verbose - return expanded command



# WHO EXEC

- Show who is logged on in an easy to view fashion with optional search
- Help:

Name: WHO EXEC - sort output of QUERY NAMES

Usage: WHO {FILTER}

Where: FILTER is an optional search pattern



# Lesser EXECs

- **QA EXEC**
  - **Runs QUERY ACCESSED**
  - **Saves a few keystrokes**
- **MKVMARC EXEC**
  - **Builds the ZVMTOOLS VMARC file**
  - **Move all EXECs to z/VM**



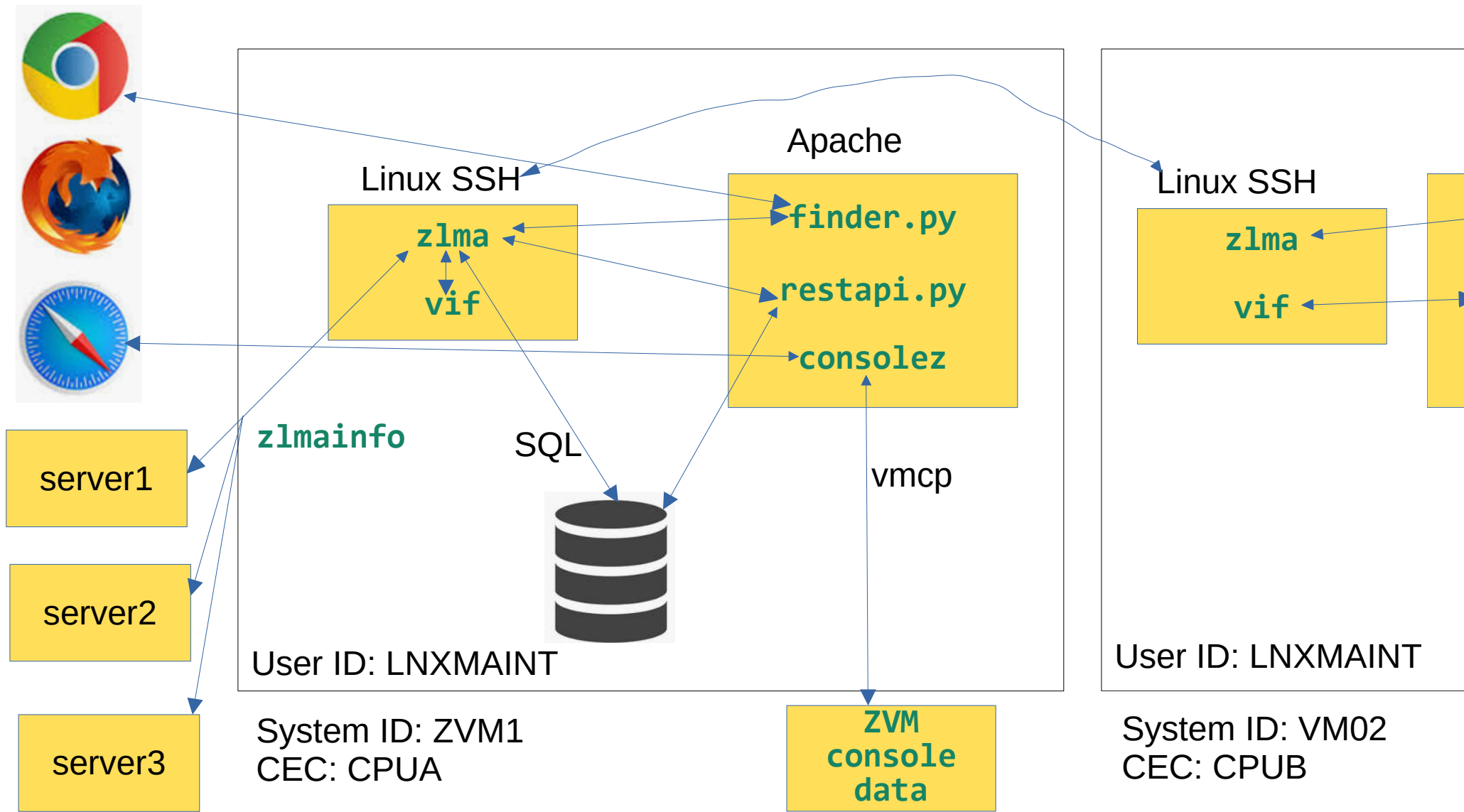
# Sources

- Where do I get the code?
  - Zvm-tools
  - <https://github.com/openmainframeproject/zvm-community-tools/tree/main/ex>
  - Zlma - <https://github.com/mike99mac/zlma>
- Questions installing/using?
  - Email: [mmacisaac@sinenomine.net](mailto:mmacisaac@sinenomine.net)

# z/ma overview

- **z/VM and Linux modern administration**
  - The problem
    - 3270/”green screen” is not being taught in University
      - New hires are not eager to learn it
    - z/VM administration has not changed much
      - Many other solutions have failed or are foundering
      - Proprietary solutions require vendor lock-in
  - The solution
    - A new free app that is:
      - Browser-centric
      - Linux command line
      - 3270 as little as possible
    - This **does not** replace the z/VM Sysad.min
      - It enables zLinux personnel to learn some z/VM and do self service

# zlma block diagram



# VIF hypervisor

- 'VIFCMS hyp collect' => problem determination data (send to IBM)
- 'VIFCMS hyp disk add image 1234'
  - Add to well known pool name?
- 'VIFCMS hyp disk add paging abcd'
- 'VIFCMS hyp errors' => hardware errors (send to IBM)
- 'VIFCMS hyp restart' => SHUTDOWN REIPL?
- 'VIFCMS hyp service' => z/VM Development?
- 'VIFCMS hyp shutdown' => SHUTDOWN?
- 'VIFCMS hyp verify' => perform z/VM consistency checks

# VIF image

- 'VIFCMS image create linux1' => clone a Linux
- 'VIFCMS image delete linux1' => no DIRM PURGE just yet
- 'VIFCMS image network linux1 add 360' => add OSA triplet - need name as an arg?
- 'VIFCMS image set linux1 storage 8'
- 'VIFCMS image set linux1 cpus 4'
- 'VIFCMS image start linux1'
- 'VIFCMS image stop linux1'
- 'VIFCMS image stopall' => loop and call 'image stop' on each

# VIF disk

- 'VIFCMS disk copy linux1 150 linux2 150'
- 'VIFCMS disk create linux1 150 2G'
- 'VIFCMS disk delete linux1 150'
- 'VIFCMS disk share linux1 151 linux2 151'

# VIF query

- 'VIFCMS query all' - all of the below
- 'VIFCMS query disks' - details on Linux disks
- 'VIFCMS query level' - Q CPLEVEL
- 'VIFCMS query network' - Linux adapters, IP addresses
- 'VIFCMS query paging' - used/available
- 'VIFCMS query performance' - overall summary
- 'VIFCMS query volumes' - image and paging volumes

# Sources

- Email: mike99mac@gmail.com
- Where do I get the code?
  - <https://github.com/openmainframeproject/zvm-community>