

# z/VM Survival Guide for the Linux Sysadmin

Daniel P. Martin  
Rocket Software, Inc.

Wednesday, 14 August 2013  
Session 13484

# About Rocket Software, Inc:

- Rocket Software is a leading global developer of software products that help corporations, government agencies and other organizations reach their technology and business goals. 1,100 Rocketeers on five continents are focused on building and delivering solutions for more than 10,000 customers and partners – and five million end users – in three key areas.
- **Rocket Intelligent Infrastructure Solutions** allow organizations to maximize the impact of their IT investment and enhance business performance with the addition of smart software that complements, manages, and optimizes the full capabilities of their IT infrastructure. Our solutions deliver value by "learning" and adapting to users' unique environments, including auto-discovering network topologies, monitoring existing backup systems, optimizing the use of storage devices, routing information based on subject matter determination, and transforming and visualizing data drawn from across the enterprise. We understand your infrastructure challenges and we understand that all your IT infrastructure is interconnected. Our solutions focus on integration and intelligence and provide value from your servers, to your applications, through your network, to your database and storage, and everything in between.
- **Rocket Mobility Solutions** make secure, focused, anywhere/anytime access to corporate data a reality through comprehensive Business Intelligence, Analytics, and Dashboarding solutions. Coupled with powerful systems for generating mobile UIs, creating powerful mobile user experiences, and managing and deploying leading-edge mobile applications, our solutions ensure that valuable corporate information is available anytime, anywhere.
- **Rocket Mainframe Solutions** enable you to deploy more existing – and yes, NEW – workload on IBM System z. Our mainframe solutions help you reduce cost and deliver more by reducing management overhead; optimizing transaction processing, delivering millions of reliable, secure transactions per day; and meeting demanding SLA's for performance/response times, uptime/availability and back-up/disaster recovery. And we can help you get even more out of your mainframe with newer technologies like zBX, IDAA, and zIIP.
- We are proud of our products, our talented Rocketeers, and the trusted relationships we have with our customers and partners around the globe.
- You can learn more about our core values in the [most recent letter](#) Andy Youniss, our President and CEO, sent to our customers and partners.

# About your speaker

- Career
  - Now:
    - Software Developer for Rocket Software since Feb, 2003
      - *Member of the development team responsible for producing IBM Archive Manager for z/VM, IBM Backup and Restore Manager for z/VM, IBM Operations Manager for z/VM and IBM Tape Manager for z/VM*
    - Current SHARE VM Project Manager
  - Then:
    - University of Arkansas – IT Security Manager, Open Systems Support Team Lead SysAdmin, VM Systems Programmer since 198mumble...
- Outside “The IT Industry”:
  - Sundry other interests; feel free to inquire.

# Linux Sysadmin: The Worst Job in the World?

**Subject: The Worst Job in the World**

**From: Michael Tiemann <tiemann@cygnus.com>**

"Being a Unix system administrator is like being a tech in a biological warfare laboratory, except that none of the substances are labeled consistently, any of the compounds are just as likely to kill you by themselves as they are when mixed with one another, and it is never clear what distinction is made between a catastrophic failure in the lab and a successful test in the field."

-- Michael Tiemann

(Then: Co-founder, Cygnus Solutions, 1989-1999)

(Now: Vice President of Open Source Affairs, Red Hat Inc.)

# Oh. Really?



*Let's discuss this in more depth...*

# WHAT DOES IT ALL MEAN!?

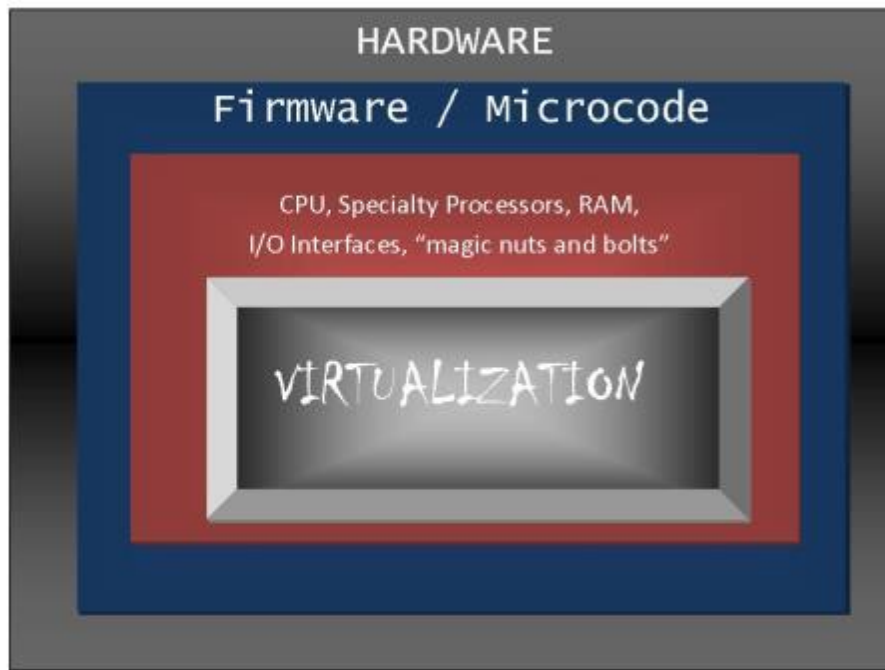
- If your professional background can produce a statement like that, you can handle **anything**.
- And now, a brief rant:
  - The word “mainframe” came in to common use in 1952.
    - *Now, we say ‘rack’. Then, they said “frame”.*
  - Anybody who thinks use of the word “mainframe” today still refers to 62-year-old technology is demonstrating a significant failure of the learning process.

# Agenda:

## Let's get down to business...

- Computer Architecture 101
- Some handy analogies:
  - Command shell
  - File systems
  - Editors
  - Pipes
  - Scripting
  - Useful z/VM tools
  - Learning resources

# Computer Architecture 101: Gross oversimplification

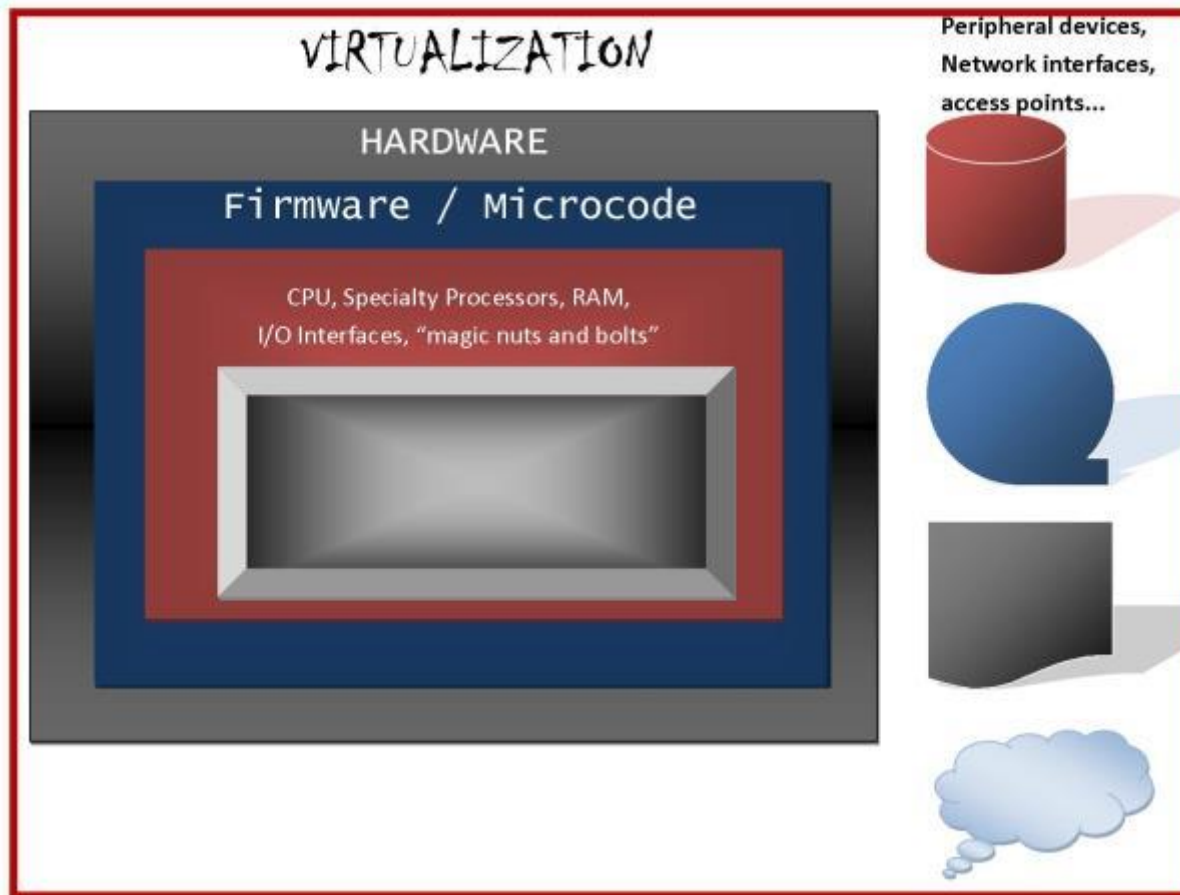


Peripheral devices,  
Network interfaces,  
access points...

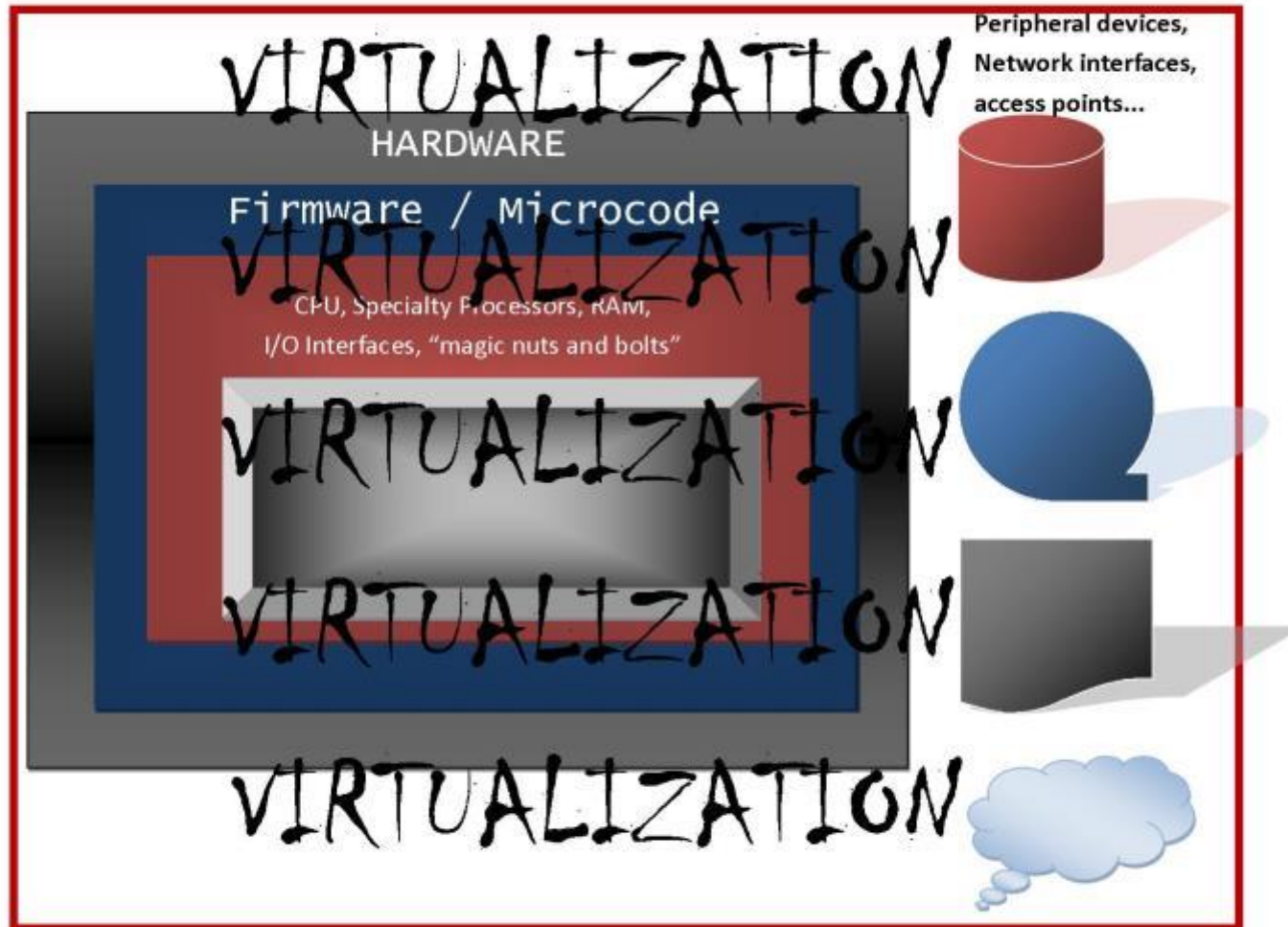




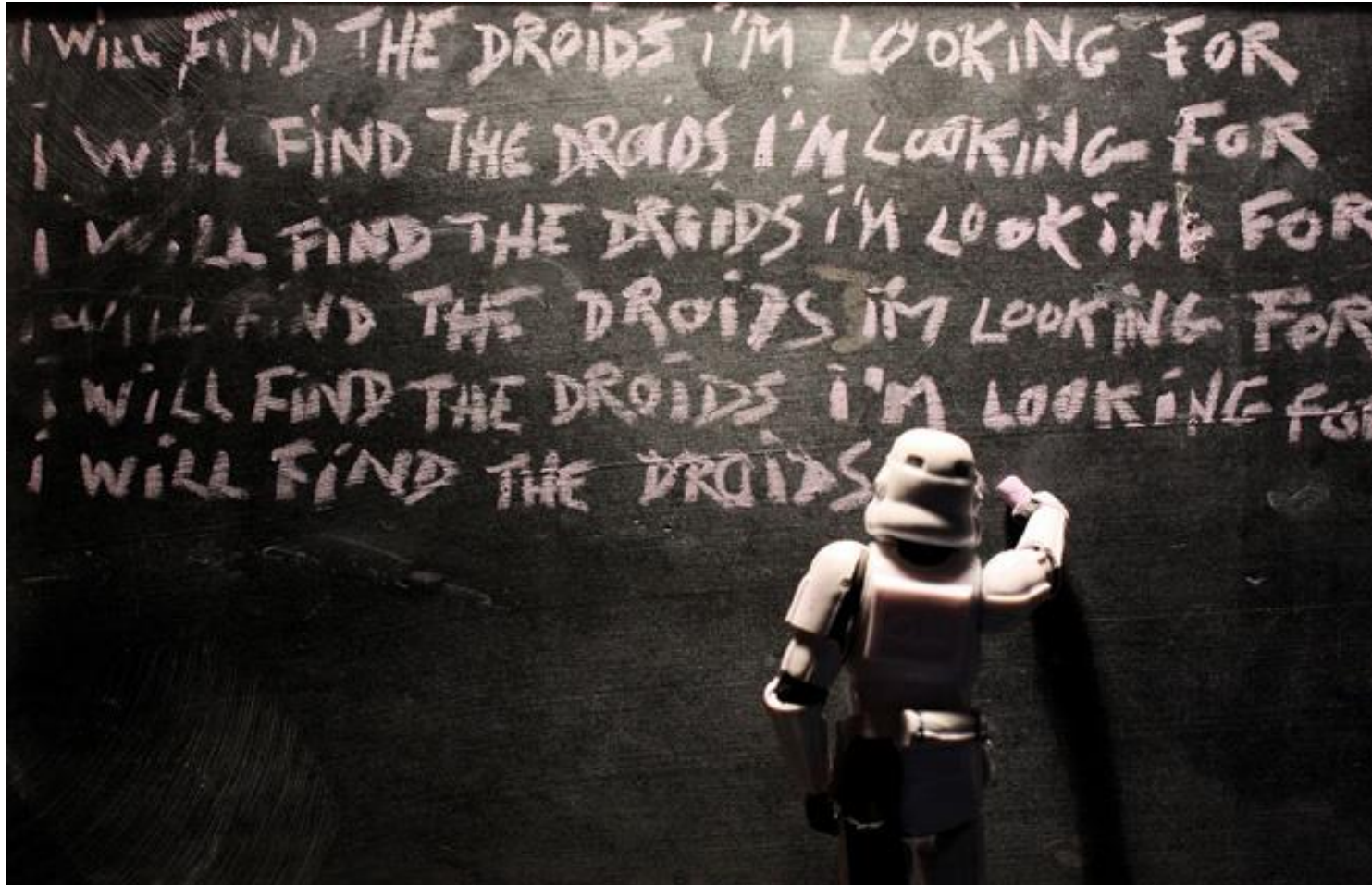
# Computer Architecture 101: Gross oversimplification



# Computer Architecture 101: Gross oversimplification



# Computer Architecture 101: Gross oversimplification



# FOUNDATION:

- Pick your tools, know your goals, know your customers
  - Better still, know your customer's needs and expectations...
- Forget, for the moment, that we're talking about a virtualized, shared-resource computing environment.
- Forget, for the moment, that we're talking about "Enterprise-class Business Server Ecosystems" (a/k/a "The Mainframe"), and let's look at a task list:

Or...

## “How to deploy a Linux System, Anywhere”



- You will need:
  - ...stuff you already know about
    - Although though the dialect on System z may differ...
  - A processor
  - Fast, volatile tightly-connected storage (“RAM”, “Memory”, “Storage”)
  - Slow, non-volatile, somewhat less tightly-connected storage (“Disk”, “Storage”, “DASD”)



Or...

## “How to deploy a Linux System, Anywhere”

- Installation media (Optical media, USB stick, tape (!?), remote FTP, something)
- Network connectivity
  - (usually... - non-connected, monolithic systems are a rarity)
- Some sort of “Human Interface Device”
  - Terminal, web browser, VNC session, *something*
  - Generically, Display and Keyboard
    - *More generally, input and output...*

## You will also need...

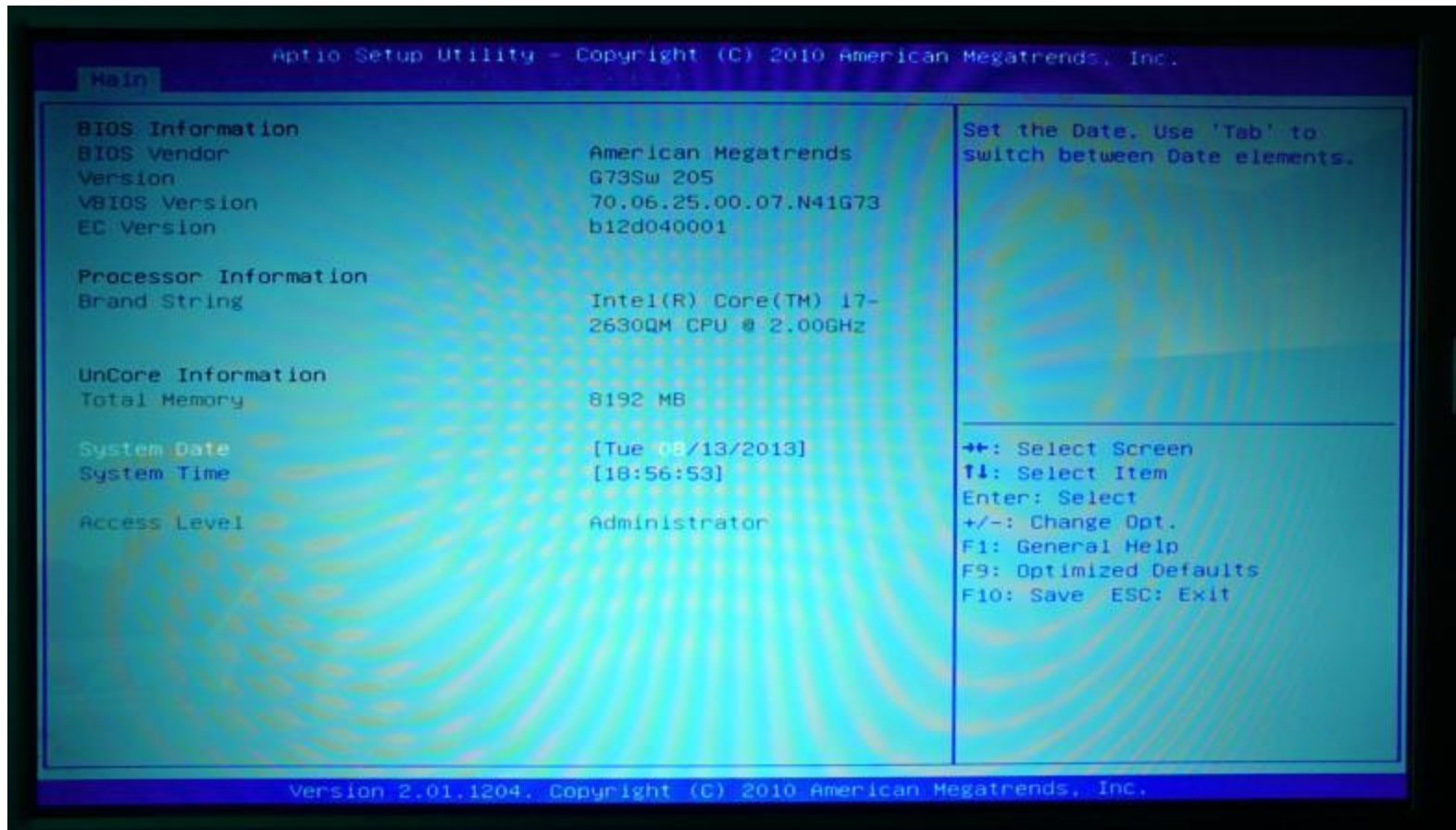
- Architecture-specific, platform-specific knowledge...
- A plan...
- And, in a perfect world, a clue...
  - ...or ready access to somebody who has a clue...
    - *Also sometimes known as “institutional knowledge”*

## Some quick examples:

- Generally speaking, “Resource Provisioning Tasks” are pain points...
  - Requisitions, Purchase Orders...
  - Delivery, un-crating, installation...
  - Electrical, networking, plumbing, cooling...
  - “Stuff” to be allocated, policies to guide allocation...
- ...but you’re in a cozy chair, with good A/C and a nice sound system, instead of a biological warfare laboratory.
  - ...probably...
    - Two words: **Career Counseling**



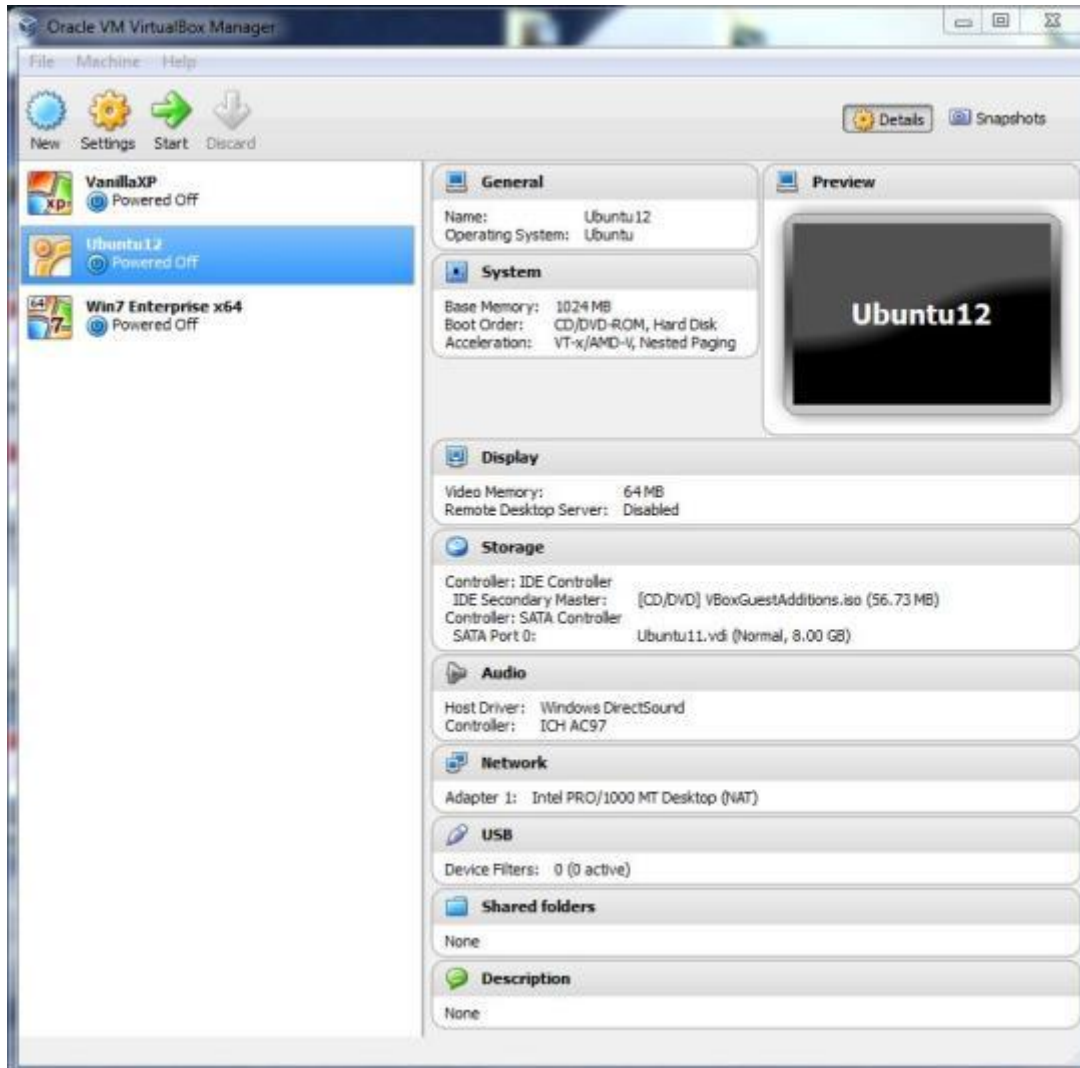
# Resource provisioning: intel platform



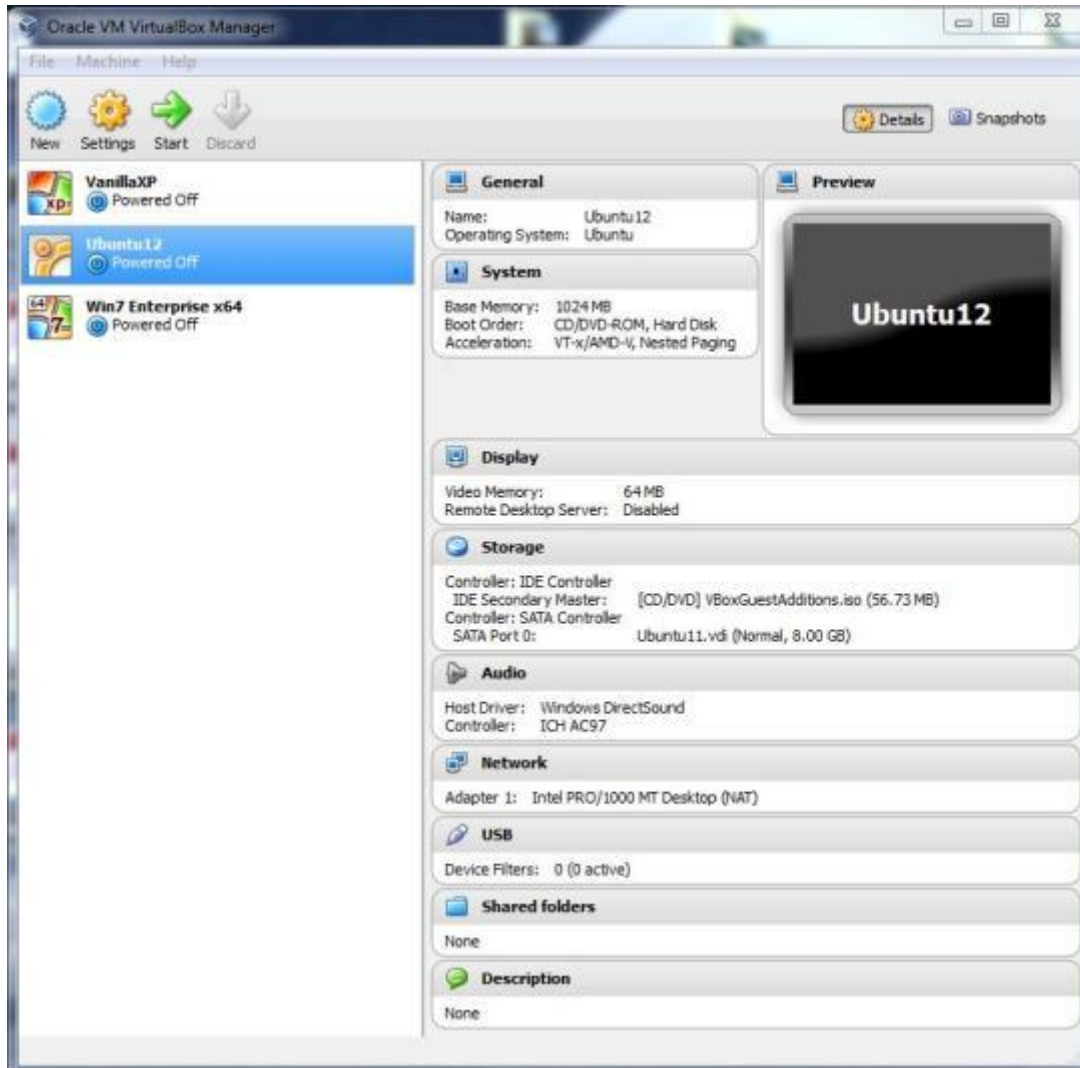
# Resource provisioning: SPARC

```
ok probe-scsi
/pci@8,600000/SUNW,qlc@2
LiD HA LUN --- Port WWN --- ----- Disk description -----
 0  0  0  2100002037bd356f SEAGATE ST318304FSUN18G 042D
 6  6  0  508002000011fd5d SUNW      SUNWGS INT FCBPL9216
 8  8  0  2100002037bd3981 SEAGATE ST318304FSUN18G 042D
```

# Resource provisioning: VirtualBox

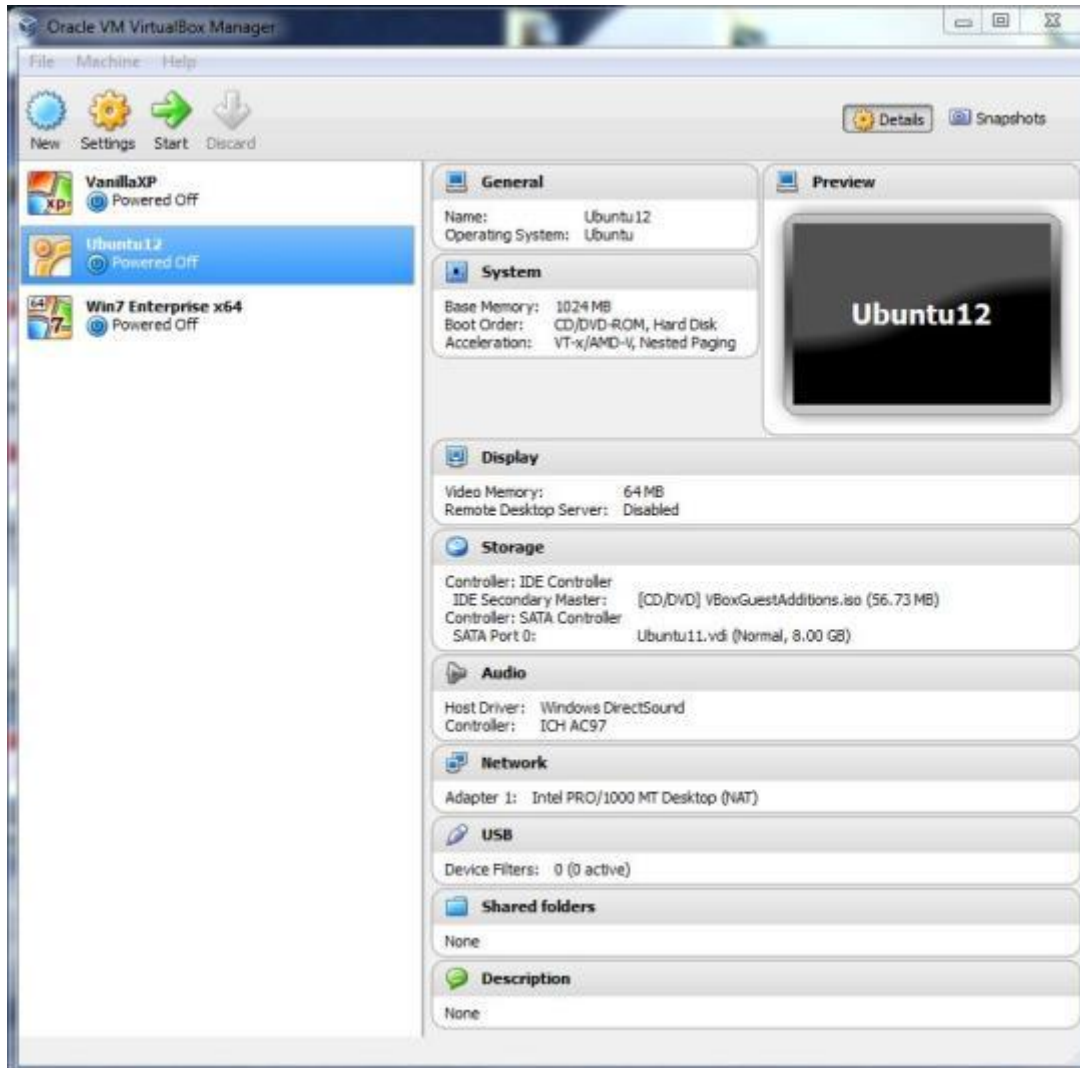


# Resource provisioning: VirtualBox



Notice how more overt virtualization just crept into the picture?

# Resource provisioning: VirtualBox



Those are virtual machine definitions...

# Resource provisioning: z/VM

```
Session C - [24 x 80]
LNXBKR  DIRECT  A1  F 80  Trunc=72 Size=15 Line=3 Col=1 Alt=0

===== * * * Top of File * * *
===== USER LNXBKR XXXXXXXX 512M 4G G
=====   IPL CMS
=====   LOGONBY ADMIN1 ADMIN2 ADMIN3
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7.>
=====   MACHINE ESA
=====   XAUTOLOG ADMIN1 ADMIN2 ADMIN3
=====   CONSOLE 0009 3215 T OPMGRM1
=====   NICDEF 9000 TYPE QDIO LAN SYSTEM VOSA1
=====   SPOOL 000C 2540 READER *
=====   SPOOL 000D 2540 PUNCH A
=====   SPOOL 000E 1403 A
=====   LINK MAINT 0190 0190 RR
=====   LINK MAINT 019E 019E RR
=====   LINK MAINT 019D 019D RR
=====   LINK RHELLNX 0191 0191 RR
=====   MDISK 0201 3390 1 10016 VMVOL9 MR READPASS WRITPASS MULTPASS
===== * * * End of File * * *
=====>

X E D I T  1 File
MA c 01/001
```

So is this...

# Resource provisioning: z/VM

```
Session C - [24 x 80]
LNXBKR  DIRECT  A1  F 80  Trunc=72 Size=15 Line=3 Col=1 Alt=0

===== * * * Top of File * * *
===== USER LNXBKR XXXXXXXX 512M 4G G
=====   IPL CMS
=====   LOGONBY ADMIN1 ADMIN2 ADMIN3
=====   |...+...1...+...2...+...3...+...4...+...5...+...6...+...7.>
=====   MACHINE ESA
=====   XAUTOLOG ADMIN1 ADMIN2 ADMIN3
=====   CONSOLE 0009 3215 T OPMGRM1
=====   NICDEF 9000 TYPE QDIO LAN SYSTEM VOSA1
=====   SPOOL 000C 2540 READER *
=====   SPOOL 000D 2540 PUNCH A
=====   SPOOL 000E 1403 A
=====   LINK MAINT 0190 0190 RR
=====   LINK MAINT 019E 019E RR
=====   LINK MAINT 019D 019D RR
=====   LINK RHELLNX 0191 0191 RR
=====   MDISK 0201 3390 1 10016 VMVOL9 MR READPASS WRITPASS MULTPASS
===== * * * End of File * * *
=====>

X E D I T  1 File
MA c 01/001
```

So is this...

...but we changed  
platforms again.

# Resource provisioning: Virtualization, z/VM-style

```
Session C - [24 x 80]
LNXBKR  DIRECT  A1  F 80  Trunc=72 Size=15 Line=3 Col=1 Alt=0

===== * * * Top of File * * *
===== USER LNXBKR XXXXXXXX 512M 4G G
=====   IPL CMS
=====   LOGONBY ADMIN1 ADMIN2 ADMIN3
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7.>
=====   MACHINE ESA
=====   XAUTOLOG ADMIN1 ADMIN2 ADMIN3
=====   CONSOLE 0009 3215 T OPMGRM1
=====   NICDEF 9000 TYPE QDIO LAN SYSTEM VOSA1
=====   SPOOL 000C 2540 READER *
=====   SPOOL 000D 2540 PUNCH A
=====   SPOOL 000E 1403 A
=====   LINK MAINT 0190 0190 RR
=====   LINK MAINT 019E 019E RR
=====   LINK MAINT 019D 019D RR
=====   LINK RHELLNX 0191 0191 RR
=====   MDISK 0201 3390 1 10016 VMVOL9 MR READPASS WRITPASS MULTPASS
===== * * * End of File * * *
=====>

X E D I T  1 File
MA c 01/001
```

So is this...

...but we changed  
platforms again.

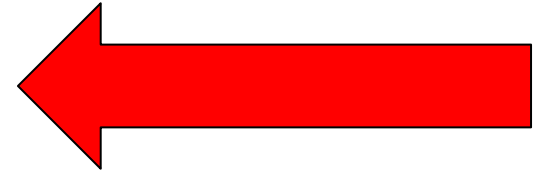
Yes, it's a green  
screen. Sorry  
about that...



# Here's the point:



# REAL



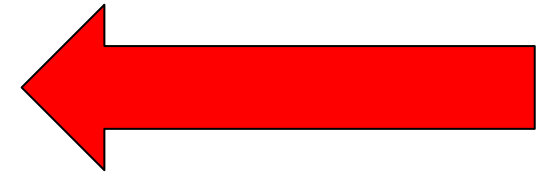
# Here's the other point:

```
Session C - [24 x 80]
LNXBKR  DIRECT  A1  F 80  Trunc=72 Size=15 Line=3 Col=1 Alt=0

===== * * * Top of File * * *
===== USER LNXBKR XXXXXXXXX 512M 4G G
=====   IPL CMS
=====   LOGONBY ADMIN1 ADMIN2 ADMIN3
=====   |...+...1...+...2...+...3...+...4...+...5...+...6...+...7.>
=====   MACHINE ESA
=====   XAUTOLOG ADMIN1 ADMIN2 ADMIN3
=====   CONSOLE 0009 3215 T OPMGRM1
=====   NICDEF 9000 TYPE QDIO LAN SYSTEM VOSA1
=====   SPOOL 000C 2540 READER *
=====   SPOOL 000D 2540 PUNCH A
=====   SPOOL 000E 1403 A
=====   LINK MAINT 0190 0190 RR
=====   LINK MAINT 019E 019E RR
=====   LINK MAINT 019D 019D RR
=====   LINK RHELLNX 0191 0191 RR
=====   MDISK 0201 3390 1 10016 VMVOL9 MR READPASS WRITPASS MULTPASS
===== * * * End of File * * *
=====>

X E D I T 1 File
MA c 01/001
```

**VIRTUAL**



# Fundamental assumptions

- Physical resources are already in place
  - “Cabling, plumbing and rigging” tasks are complete
    - Your shiny new zEnterprise BC12 is installed
      - *Or your new laptop is out of the packing case...*
      - *...or the curb outside is covered in empty Newegg boxes...*
      - *...or your raised floor is covered with racks...*
        - *...except for the parts covered by power & cooling gear...*

# Fundamental assumptions

- You know a thing or two about Linux already...

# Fundamental assumptions

- You know a thing or two about Linux already...
- You (or a Subject Matter Expert) have an installed, running z/VM operating system...

# Fundamental assumptions

- You know a thing or two about Linux already...
- You (or a Subject Matter Expert) have an installed, running z/VM operating system...
- You've found motivation to learn more about z/VM.

# And so, a short set of analogies...

a·nal·o·gy  
/əˈnaləjē/

Noun

1. A comparison between two things, typically on the basis of their structure and for the purpose of explanation or clarification.
2. A correspondence or partial similarity.

# Analogies: Command Shell

## Linux Environment

- Many flavors
  - sh, ksh, tcsh, bash
- Runs as a process
  - Kernel provides services
- Commands are processes
  - ...or native shell functions
- Boolean privilege scheme
  - ...generally
    - ...*subject to audit mechanism*
- Line-mode, full-duplex
  - ...vt100 or similar

## z/VM Environment

- One flavor
  - The CMS operating system
- Runs in a Virtual Machine
  - CP provides services
- Commands are programs
  - ...or native CP commands
- Multiple privilege levels
  - For CP functions
    - ...*and some CMS*
- Block-mode 3270 terminal
  - ...3270 emulator



# Notes:

## The z/VM Virtual Machine Terminal

- Linux (and most of the rest of the world) say “terminal”
- System z people often use the word “console”
- Library tip: z/VM V6R1.0 CMS Primer (SC24-6172)
  - “If you’re not sure where to start, start here.”

# Detail: The z/VM Virtual Machine Terminal

- Some people call it a console



# Detail: The z/VM Virtual Machine Terminal

- Some people call it a console



Please note:

This is NOT a  
zEnterprise BC12

# Detail: The z/VM Virtual Machine Terminal

- Some people call it a console

Hint:

That's a phone. It's attached to a wall by a "wire".



# Detail: The z/VM Virtual Machine Terminal

- Some people call it a console

Hint:

That's a phone. It's attached to a wall by a "wire".



It was the 1970's. We left that decade for lots of good reasons.

# Analogies: Command Shell

- Dual personality nature of the z/VM “command shell”
  - CP commands
    - ...belong to the hypervisor
    - ...are NOT part of the operating system executing in the virtual machine
    - ...are present regardless of what (if any) OS is executing
    - ...resemble firmware / BIOS interaction with other platforms in many ways. They generally display or alter the Virtual Machine state.
  - ...are subject to a fine-grained permission structure
    - *Built-in “native” CP privilege class structure*
    - *ESM (External Security Manager) control*
  - (Arguably) Provides best-of-breed machine language debugging
    - *No arguments today, please...*

# Analogies: Command Shell

- Dual personality nature of the z/VM “command shell”
  - CMS commands
    - ...are NOT part of the CP hypervisor
      - *...but can invoke CP commands or services*
    - ARE part of the Conversational Monitor System OS
    - ...base set consists of the IBM “baked in” CMS commands
    - ...can be independent programs executed via CMS
      - *...can be part of the z/VM base*
      - *...can be part of IBM add-on software (e.g. DIRMAINT)*
      - *...can be part of OEM software*
      - *...or can be “home grown”*
    - CMS is an operating system. So is z/OS, z/VSE and z/TPF. So is Linux for System z. So is the CP hypervisor.
      - **All can** *execute in a virtual machine. CMS runs **only** in one.*

# Analogies: File Systems

## Linux Environment

- Many flavors
  - ext2, ext3, jfs, btrfs...
- Plugs into abstraction layer
  - POSIX calls
- Reside on a variety of media
- 3-tier privilege scheme
  - ...generally
    - ...*subject to fs details*
- Stream-oriented I/O
  - ...almost always

## z/VM Environment

- Flavor varies by guest OS
  - CMS: Minidisk (EDF) or SFS
- No abstraction layer in CMS
  - Exception: POSIX in OE
- Disk (“DASD”) resident
- Access control not like Linux
  - CP control of MDISK access
  - Fine-grained SFS control
- Record / block-oriented I/O
  - ...generally...
  - Dodging SAN for z/VM here



# Analogies: File Systems

- CMS minidisk file systems
  - Are at the core of the z/VM installation and service process
  - DO NOT SHARE WELL WITH OTHERS
    - One writer, multiple readers: do-able, with some risk of “fuzz”
    - Multiple writers: Chaos and destruction will ensue
      - *NO serialization mechanism in the CMS EDF file system!*
  - A bit “odd” relative to other operating systems
    - Name space management
      - *Filename filetype fm*
      - *Flat, not hierarchical*
  - Access control is managed via CP oversight of virtual device
    - “File Mode 0” is *\*not\** a security mechanism!

# Analogies: File Systems

- CMS Shared File System (SFS)
  - Can be used as the repository for the z/VM installation and service process
  - Shares VERY well with others, monolithic or clustered.
    - LUW mechanism ensures consistent view from open to close
    - Multiple writers: 2<sup>nd</sup> writer is warned, but can force the issue.
      - *Last one to write wins. “Winning” is a value judgment...*
  - A bit less “odd” relative to other operating systems
    - Name space management
      - *Filename filetype fm*
      - *Hierarchical, sort of – z/OS-like directory qualifiers*
  - Access control is managed via SFS or ESM; is auditable
    - “File Mode 0” is just another file mode number.

# Analogies: Editors

## Linux Environment

- Many options
  - emacs, vi / vim, pico, sed...
- Interacts with files like any other application
- Each editor has its own fan base
  - ...well, maybe not 'sed'...
- Generally “full screen” oriented ('curses' lib)
  - ...generally...

## z/VM Environment

- One primary option: XEDIT
  - ...but there are others
- Interacts with CMS files like any other CMS application
- Each editor has its own fan base
  - ...well, maybe not VM ISPF...
- Full-screen oriented
  - Nature of 3270 terminal
  - Can still be forced to line mode

# Analogies: Editors

- XEDIT is a powerful, flexible, robust, highly programmable editor. Arguably, it's the emacs of z/VM - CMS
  - Scripted control with REXX and PIPEs
  - Can be “re-skinned” to function as a 3270 dialogue manager
  - Interaction is largely confined to CMS-native files
    - ...with a few weird exceptions...
    - ...including CMS / OE
  - Like emacs, lots and lots of Raving Fans
    - STILL not here for an argument today.
  - “Clones” available for Unix/Linux environments:
    - ‘the’ – The Hessling Editor (open source)
    - uniXEDIT – commercial (The Workstation Group)

# Analogies: Scripting Languages

## Linux Environment

- Many, many options
  - shell, perl, php, others...
- Interpreted, not compiled
- Part of Linux/Unix shell environment
- Generally portable from one release to the next
- Arguably portable from one distro to another
  - ...so is C, right?

## z/VM Environment

- One primary option: REXX
  - ...other options exist
- Interpreted or “pseudo-compiled”
- Part of the CMS operating system
- Highly portable from one release of z/VM to the next
- Arguably portable
  - ...so, there’s an ANSI standard, right?

# Analogies: Scripting Languages

- REXX is the scripting language of choice for CMS.
  - Fossilized code written in predecessor scripting languages, EXEC and EXEC-2, may be found lurking in dark corners of parts of long-standing VM shops.
  - Two schools of thought:
    - “If it ain’t broke, don’t fix it.”
    - “Terminate with extreme prejudice.”
    - This is a judgment call; EXEC and EXEC-2 are still supported; unless there’s incentive to re-write applications, should you?.
- Yes, there is an ANSI standard (see [www.rexxla.org](http://www.rexxla.org))
- Unix / Linux ports:
  - ‘regina’ – (open source – Mark Hessling, of ‘the’ fame)
  - uniREXX – commercial (The Workstation Group).
- ...and more. See [www.rexxla.org](http://www.rexxla.org) for details.

# Analogies: Pipes

## Linux Environment

- “Baked into” shell environment
- Native function; glues disparate shell commands and applications together
- Portability depends on commands in the pipe expression
- Fantastically powerful stuff.

## z/VM Environment

- Part of CMS
  - Can call CP services
- Native CMS application; glues disparate CMS commands and PIPE functions together
- Portability depends...
- Modeled on native Unix/Linux pipe function
- Fantastically powerful stuff.

# Analogies: Pipes



- CMS PIPEs are like Linux / Unix pipe expressions in many ways
  - Utilization varies from simple command output redirection to highly complex service virtual machine (that's "daemon" in Linux-ese) processes.
- As in Linux, can be invoked from within a script or from the command prompt.
- The amazing thing about CMS PIPE is that we did so much in z/VM, for so long, without it.
- Tremendous power in a very small package
- If you spend more than the bare minimum of time in CMS, learn to use PIPE.





## Closing thoughts: Useful z/VM Tools

- Just like the Linux / Unix community, there are a number of “open source-like” tools available in the z/VM community.
  - TRACK – ‘top’ for z/VM, on steroids, with chocolate sprinkles
    - Seriously, every z/VM SysProg needs this
    - <http://vm.marist.edu/track/>
  - VMARC – ‘zip’ for CMS; extremely useful for archiving files, coincidentally useful for shipping CMS files via e-mail
  - <http://www.vm.ibm.com/download> for VMARC and many other useful CMS tools.
- Ask the networking contacts you’ve developed at SHARE!!

# Closing thoughts: Learning resources

- There are active support communities for z/VM and for Linux on System z
  - *You're there right now.*
- Web / e-mail resources
  - [listserv.uark.edu/archives/ibmvm.html](http://listserv.uark.edu/archives/ibmvm.html) – The IBMVM mailing list
  - [www2.marist.edu/htbin/wlvindex?linux-390](http://www2.marist.edu/htbin/wlvindex?linux-390) – The Linux-390 mailing list
- User Groups
  - SHARE, WAVV, The VM Workshop, and regional user groups

# Closing thoughts: Learning resources

- IBM Resources
  - [www.vm.ibm.com](http://www.vm.ibm.com) – IBM portal to all things z/VM-related
  - [www.vm.ibm.com/library](http://www.vm.ibm.com/library) - z/VM Documentation Library
- Linux on System z – Branded distro resources
  - Red Hat: [www.redhat.com/products/enterprise-linux/for-ibm-system-z/](http://www.redhat.com/products/enterprise-linux/for-ibm-system-z/)
  - SuSE: [www.suse.com/products/systemz/](http://www.suse.com/products/systemz/)
- Other resources:
  - You're here. Other people are here. Ask around!

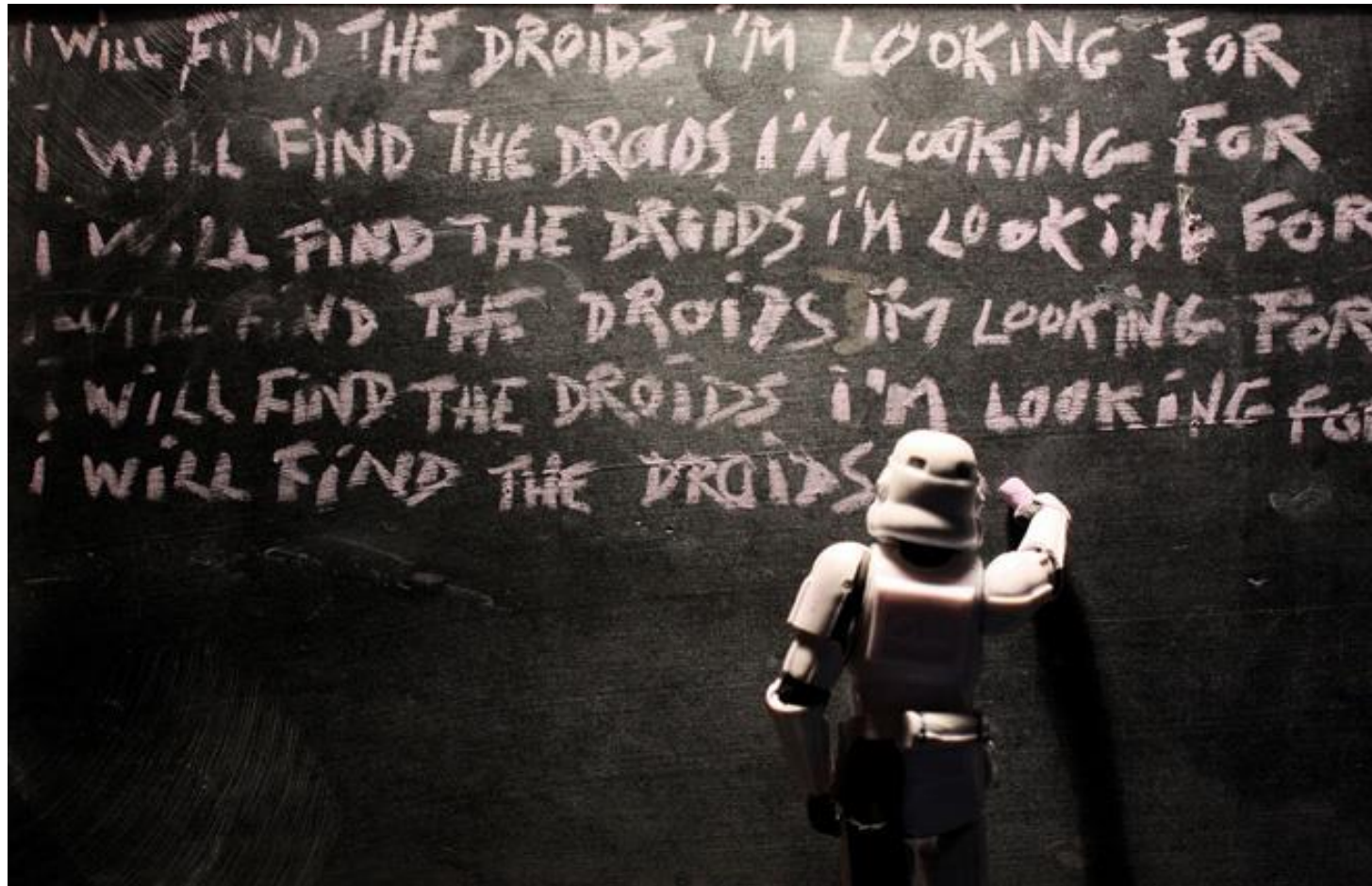
## Closing thoughts: No, really...

- Linux (or Unix) on intel was hard when you started there.
- Unix (or Linux) on <architecture of choice> was hard...
- The z/VM Control Program (CP) is a System z operating system. It does just one thing:
  - Virtualize the System z architecture.
  - It does this very, very well.
    - “Virtualization, before virtualization was cool.”

# Closing thoughts: No, really... Not kidding this time...

- All architectures have a learning curve.
- All virtualization technologies have a learning curve.
- “Hard” is a subjective value judgment.
- “Different” is a word which means “something I haven’t learned all about... yet.”

# Final comments & discussion?



# Thank you!

Time is really the only capital that any human being has and the thing that he can least afford to waste or lose.

*– Thomas Edison*

## Thank you for choosing to spend this time with me today!

Daniel P. Martin – Senior Software Developer  
Rocket Software, Inc.  
dan.martin@rocketsoftware.com