Story of A Linux Box

From Hardware to High Performance Computing

By Kshitij Sudan

IVth Year COE
What we’ll talk about today!!

• Modern computer architecture and it's relation to Linux.
• Basics of a Linux system.
• Installing Linux.
• Advance Linux concepts.
• Adaptability of Linux and LAMP suite.
• Suitability of Linux to specific applications, in particular high performance computing.
Computer Architecture? What?

• Computer architecture is **NOT** about using computers to design buildings.

• **Computer Architecture** is the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals - focusing largely on the way by which the [central processing unit](#) (CPU) performs internally and accesses [addresses in memory](#).

OK I agree that was geeky.........

Contd.
Computer Architecture? What?
(Cont.)

• Simply put, it’s *the physical design and realization of a computer*.

• Even more simply, it’s the *hardware aspect of a computer*.

• Includes the *microprocessor*, physical memory (*RAM*), secondary storage (*HDD*), peripheral interconnects (*Printers, USB etc*), I/O devices (*Keyboard, NIC’s*) and their interconnects (the *motherboard*).
Computer Architecture? What?

The Microprocessor

• It’s the central processing unit (CPU) of a computer.

• Integrates over half-a-billion transistors on a thumbnail.

• Well over 40 different architectures and over 1000 different CPU’s designed till today.
Computer Architecture? What?  

**The Microprocessor**

- **A CPU typically contains:**
  
  > Digital logic implementing adders, shifters, floating point units etc. **(ALU)**
  > Logic to **fetch and decode** instructions.
  > Logic for **control** and performance improvement like **branch predictors, BTB’s** etc.

Contd.
Computer Architecture? What?

The Microprocessor

> Registers to store temporary values.
> L1 and L2 cache.

….. besides other components depending upon the specific use of the chip.

And now a feast for the eyes............
IBM PowerPC 601 die
AMD Am29k and x86 Dies
Intel Pentium
Cutting Edge in Microprocessors

• Application specific processors, like ASICs, microcontrollers, network processors etc.
• FPGA based Processors
• Symmetric Multi Processors
• Chip Multi Processors
• Re-configurable Processors
What has all of this to do with Linux?

• Underlying **hardware defines the services and capabilities the OS can provide.**

• Hardware and software co-habitat in a system, hence **not exclusive of each other.**

• **Hardware-Software co-design improves performance**, notably in application specific system design like network processors.
So how’s Linux better than others on these criterion?

• You have the source, you change it to accommodate your needs.
• Absolutely free, you just need to be smart enough to adopt and use it! *(Are you?)*
• Great support with many like-minded people working on similar problem as yours all over the world.
• Already ported to embedded platforms and a big hit in server space.
I am still not convinced why I should adopt Linux!

• **Greatest learning opportunity** in college (my humble opinion of-course!).
• **A new perspective on living life** (Yes! Linux is spiritual too………)
• **Getting to know smart people** and making great friends.
• **Above all**, the satisfaction of making something work with you own hands.

Are you convinced now?
OK, so what is this Linux anyway?

- An **Open Source** implementation of the UNIX OS, which **runs on many different hardware** platforms including Sparc, PowerPC, Alpha, and Intel Processors.
- Started out as a clone to the proprietary AT&T UNIX operating system.
OK, so what is this Linux anyway?

(Contd.)

• Evolved to an open source, independent operating system.
• Dominates some crucial areas of computing like embedded, server, high performance computing, research etc.
And who did this all?

- Started with **Linus Torvalds**, a student from Finland, who wrote the kernel and **Richard M Stallman**, then with MIT, who founded the free software movement and the GNU Project among other pioneers.
When?

Released on Oct. 5, 1991 by Linus, then at University of Helsinki
OK, I want in! How do I go about it?

• Learn by doing.
• Install Linux on your Desktop/Laptop and USE IT!!
• Join some USENET group which discusses topics related to Linux (comp.os.linux.* for example).
• There are no age barriers, no price controls, and no proprietary knowledge. The limitations are your own.
Fair enough! But how do I install Linux?

READ THE MANUAL
What? That sucks............
and this workshop was a waste of my time......

• No seriously, that’s the best way to learn Linux, by doing it!!

• But if you insist, here’s the algorithm to install Linux.
Algorithm for Installing Linux
a.k.a. 6 Steps to Nirvana

• STEP 1: Get a computer and Linux distro.
• STEP 2: Make Google your best friend.
• STEP 3: Find the manual for your distro using your new-found friend.
• STEP 4: READ THE MANUAL
• STEP 5: Follow the instructions in the manual and install, if installation fails, go back to STEP 4.
• STEP 6: Stop when you have a working installation.
OK, I get it. I’ll read the dammed manual and install.

So what’s next?

- Once you get comfortable working with Linux and other open source software, you delve deeper into this bottomless sea.
Advanced Linux Concepts, huh?

- The **Linux OS design is modular**.
- **Kernel** is responsible for providing most of the functionality of the OS.
- Provides services like CPU scheduler, memory management, file system, programming interfaces etc.
- It **does not include all the drivers** for the hardware but generally loads them when required.
What does the Kernel not do?

• Your favorite game is NOT the kernel.
• Does not provide any user functionality like text editor, graphics, multimedia etc.
• You never talk to the kernel directly, there’s always an interface between you and the kernel, normally it’s the shell or the GUI.
So how does it all gel together?
Some more feast for the eyes…

but this time for a geeks eye :)

Note the Background of the pic!
Linux Kernel Graphing Project

Linux Kernel v2.4.0
What was that?....?

• It was a map of the Linux kernel (v2.4.0)
• It maps the various functions of the kernel to a set of concentric rings.
• The inner most ring contain maps of kernel/, ipc/, mm/ and init/.
• The next contains code maps from fs/ and net/.
• The third ring has one segment per architecture, and the final ring has all the drivers piled together.
Have a look once more……

*Linux Kernel v2.4.0*
And now let’s watch a small movie!
3D animation of Linux source code development
(c) Feb 2001 Pascal.Brisset@wanadoo.fr
That was a 3D Rendering of the Linux Kernel

- The images are produced when the dependencies in the Linux kernel source code are depicted following a set of parameters:
  - Grey boxes represent files.
  - The green tree is the directory structure. The two main hubs are "fs/" and "net/".
  - Blue lines are function dependencies.
  - Red lines are variable dependencies.
  - Yellow flashes show file size modifications.
  - Green flashes show files being moved across directories.
  - Red flashes show new files.
Have a look once more!
3D animation of Linux source code development (c) Feb 2001 Pascal.Brisset@wanadoo.fr
Adaptability of Linux

• It’s open, hence highly configurable
• No barriers to limit of your thought, do what you want with it!
• Don’t re-invent the wheel, build on what others have already accomplished.

May-the-force-be-with-you!
LAMP, What?

- LAMP = Linux Apache MySQL PHP
- Widely popular suit used for servers on the Internet.
- Translates to **ZERO** software license costs.
- Highly robust and efficient.
- Suitable for **High Availability Computing**.
What do I do with LAMP?

• Build your own web-server at home.
• Great skill set for job market.
• Learn the 4 most important areas of computer science
  • Operating Systems (Linux)
  • Client-server computing model (Apache)
  • Databases (MySQL)
  • Programming (PHP)
But I want to do something more *geeky and cool*!!

Welcome to

High Performance Computing (HPC)
Whoa! What’s that?

• Using computers for number crunching applications.
• Applies to simulation, modeling, scientific computing, weather forecasting, engineering applications and lots more…
• Includes building the infrastructure, designing applications and much much more……
Wikipedia on HPC

• The term high performance computing (HPC) refers to the use of (parallel) supercomputers and computer clusters, that is, computing systems comprised of multiple (usually mass-produced) processors linked together in a single system with commercially available interconnects.

• Because of their flexibility, power, and relatively low cost, HPC systems increasingly dominate the world of supercomputing.
What can I do in the HPC domain?

Anything and Everything !!
How do I go about it?

• Choose a problem which needs number crunching.
• Read up all you can about the methods to solve/past work etc.
• Think of something clever to solve it more efficiently.

• JUST DO IT (Nike might sue me for this!!)
But it sounds expensive!!

• It’s not!
• Cluster computing, which is a model of HPC, employs a bunch of desktops, interconnected by an inexpensive network, working in tandem to act as a high performance system.
• Don’t be bogged down by all the numbers (let the computer be worried about that!), you think hard and stay motivated!
Moral of the story?

DO IT NOW !!
Questions?
That’s all for today!

Have a nice day 😊