z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM Software Group

TPF Users Group Fall 2005

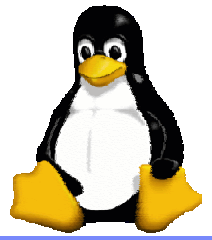
An Introduction to Linux and Linux for z/TPF

Name: Dan Weber
Venue: Ongoing TPF Education

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

The Beginning - There was once a student…

- It all started with the Finnish University Student Linus Torvalds. While he was a student at the University of Helsinki he used a Unix inspired operating system called Minix, which was mainly used for teaching purposes.
- Torvalds was not much impressed with Minix and its lack of functionality. He decided that he would start a project on his own.
- In August of 1991 Torvalds sent out an e-mail to the Usenet community, announcing his work on “Linux”
- In September of 1991 the first fragments of the Linux kernel, version 0.01, were released on the Internet. The FTP administrator created a directory called “Linux” for Torvalds to store his code and the name has stuck ever since.
- Quickly afterwards, in October 1991, the next version of the Linux kernel, 0.02, was released and was gaining more and more interest amongst developers around the world and they started to write patches, extensions, and modules for the Linux kernel and submitting them to Torvalds.
- In December 1991 version 0.11 was released. This was the first release that was self contained, meaning it was possible to compile Linux 0.11 under Linux 0.11.
- In March 1992 version 0.95 was released, which was capable of running the X Window system.
- In March 1994 the Linux kernel, version 1.0 was released
- In May 1996 the penguin mascot “Tux” appeared →



The Beginning - The Usenet Announcement

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu)
for 386(486) AT clones. This has been brewing since april, and is starting to get ready.
I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that
I'll get something practical within a few months, and I'd like to know what features most
people would want.  Any suggestions are welcome, but I won't promise I'll implement them :-)

               Linus (torvalds@kruuna.helsinki.fi)

PS.  Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable
(uses 386 task switching etc), and it probably never will support anything other than AT-
harddisks, as that's all I have :-(.
```

The Beginning - The GNU Connection

- The evolution of Linux is tightly coupled with the GNU project, which was started in 1983 by Richard Stallman
- Richard Stallman believes that good software should be free and available to all and that was not the case with Unix and the associated tools
- As a result, the purpose of the GNU project was to build a freely available Unix like operating system.
- By the time Torvalds released the first Linux kernel in 1991 the GNU project had successfully created various free Unix based development tools, utilities and applications, but it was lacking a kernel

The perfect match

Torvalds only has a kernel (Linux) and Stallman only has the tools (GNU)

- Following the initial release of Linux, Torvalds and the development community worked hard to adapt the Linux kernel to work with the GNU tools and eventually created a complete operating system, GNU/Linux
- Eventually during the 1990's Torvalds was releasing the Linux source code under the GPL (General Public License)

BSD is not Linux

There is another, very similar to Linux, Unix look-alike. It's called BSD (Berkley Software Distribution) and was originally created around 1977 by the University of California in Berkley.

Over time a few open source derivatives of BSD have been born and released:

FreeBSD	Claimed to be the "friendliest" BSD with great performance.
Darwin	Darwin is the base upon which Apple implemented its OS X operating system. It incorporates components from various BSD derivatives but mostly from FreeBSD
NetBSD	A BSD variant with the most supported hardware platforms
OpenBSD	Focused on security and cryptography support, very portable as well
SunOS/Solaris	Even the commercially quite successful SunOS/Solaris operating system has its roots in the original BSD world

BSD operating systems are often mentioned alongside GNU/Linux because they share similar characteristics (a Unix core) but they are really quite different. BSD was truly a descendant of the original Bell Labs Unix operating system while GNU/Linux was developed completely in isolation. GNU/Linux and BSD are also released under different licenses.

The Kernel - The *True* Linux

- Usually when the term “Linux” is used the entire operating system is meant. However, “Linux” only really refers to the kernel of the operating system.
- GNU/Linux operating systems are the most widely used incarnations found and consist of the combination of the GNU Operating System and the Linux kernel.
- The Free Software Foundation strongly urges everybody to use **GNU/Linux** or **Linux based GNU system** as name instead of just “Linux”. This has caused, and still does cause, a lot of confusion. Torvalds thinks that demanding to call Linux “GNU/Linux” is “just ridiculous”.

What The Kernel Does

- A kernel is the core of an operating system
- A kernel acts as an abstraction layer that provides secure and controlled access to the available hardware resources
- A kernel also controls access between computer programs (processes)
- A monolithic kernel, such as Linux, is very efficient because of the tight internal coupling of its components
- An important part of the kernel is the scheduler which controls how long a specific process should run before it has to give up control to another process. This enables the peaceful coexistence of multiple processes and allows them to run seemingly in parallel.
- Linux is built as a monolithic kernel and defines a high level interface to the underlying hardware. Because monolithic kernels can grow quite large Linux has enabled the use of loadable modules to control its growth. By using loadable modules the kernel can load and execute just the required code for a given hardware configuration and keep the resource consumption (mostly memory) to a minimum.

Kernel Versioning

Since the beginning the Linux kernel had an even/odd versioning schema:

Odd Versions (i.e. 2.5.xxx)	Development version. New functionality is added and tested
Even Versions (i.e. 2.4.xxx)	Stable version. These versions are meant for production use and only bug fixes are being applied, no new functionality.

With the release of Linux kernel 2.6 this has now changed, for now.

Because fixes and additions had to be tracked in two different branches (the odd development branch and the even production branch) some quality issues have come up. For the time being it was decided that new enhancements and features would be rolled into the current 2.6 Linux kernel at about quarterly intervals. This should streamline the development and release cycles.

There is still the possibility that a 2.7 development kernel might be created in the future, if a fundamental enough change justifies it. It's pretty much up to Torvalds as he still controls the Linux kernel releases.

Kernel Versioning

For now a four digit versioning system as been adopted.

a.b.c.d

a The kernel version, currently 2 (1.0 in 1994, 2.0 in 1996)

b Major revision of the kernel, currently 6

c Minor revision of the kernel.

For versions 2.6.10 and earlier:

This number was incremented when security patches, fixes and new drivers where added

For versions 2.6.11 and later:

It now only gets incremented when new drivers or features are being released.

d Starting with kernel 2.6.11 this forth number was introduced and only bug fixes and security patches cause this number to increment. Bigger changes will cause the minor revision (c) to increment

The most current kernel release at the moment is
>> 2.6.13.4 <<

Kernel History

Version 1.0 of March 1994 supported only single-processor i386 machines.

Version 1.2 of March 1995 added support for Alpha, Sparc and Mips processors.

Version 2.0 of June 1996 added support for more processors and included SMP support.

Version 2.2 of January 1999 added support for even more processors

Version 2.4.0 of January 2001

- CPU support: Hewlett-Packard's PA-RISC processor, Axis Communications' ETRAX CRIS ("Code Reduced Instruction Set") processors
- added ISA Plug-and-Play
- added USB and PC Card support
- added Bluetooth support (2.4.6)
- Filesystem and data storage
  + added Logical Volume Manager version 1 (LVM)
  + support for RAID devices
  + Support for InterMezzo filesystem were added. (2.4.15)

Kernel History

Version 2.6 of December 17, 2003

- integrated uClinux (for microcontrollers)
- CPU support:
  + Hitachi's H8/300 series, the NEC v850, and Motorola's embedded m68k processors
  + NUMA support, support for NCR's Voyager architecture, support for Intel's hyperthreading and PAE
- OS support:
  + Improved APIC support.
  + Increased the maximum number of users and groups from 65,000 to over 4 billion.
  + Increased the maximum number of process ids from 32,000 to 1 billion.
  + Increased the maximum number of device types (major device) from 255 to 4095
  + Increased the maximum number of devices of each type (minor device) from 255 to more than a million.
  + Improved 64-bit support and filesystems of up to 16 terabytes on common hardware.
  + Improvements to the "overall responsiveness" for interactive processes (kernel became fully preemptible).
  + A rewrite of threading infrastructure to allow the Native POSIX Thread Library (NPTL) to be used.
  + An improved module loader.
  + User Mode Linux integration.
  + Infiniband support (2.6.11)
- Storage Support:
  + LVM updated to version 2
  + support for SGI's XFS filesystem.
  + A new "system filesystem" called sysfs, destined to relieve procfs of its system related information.
  + iSCSI support (2.16.12)
  + inotify support (2.16.13)

Kernel 2.6 - A Major Improvement

With Linux kernel version 2.6 some major improvements have been made:

- Increase of internal limits (users/groups, processIDs, filesystem, devices)
- Much improved 64bit support
- More efficient interrupt handling
- Support for the Native POSIX Thread Library (NPTL)
- Handles larger symmetric multi processor (SMP) systems and Non-Uniform Memory Access (NUMA) systems
- Hyperthreading support
- Inclusion of the µClinux functionality to allow kernel to scale down for embedded systems
- Subarchitecture support, removing processor type <-> hardware platform relationship
- Better performance and throughput for Storage Area Networks (SANs)
- More secure through SELinux additions
- Enablement and support of the upcoming Xen virtualization platform
- User mode Linux

Kernel Maintenance

Linus Torvalds is usually only concerned with the maintenance of the most current kernel release and controls the release and content of new kernel releases. Therefore the maintenance oversight of older releases have been delegated.

Kernel Release	Maintainer
2.0	David Weinehall
2.2	Marc-Christian Petersen (formerly Alan Cox)
2.4	Marcelo Tosatti
2.6	Linus Torvalds

The one and only place for all Linux kernel questions and repository of all Linux kernel versions

www.kernel.org

Linux Distributions

Problem:

- Linux kernel, GNU tools, and additional components and applications are all separate packages
- User needs to obtain individual packages and create a working system
- Time consuming, error prone and maintenance intensive

Solution:

- Use one of the prepackaged Linux distributions.
- Essentially a community group or company has gone through the effort to integrate and package a complete system, built from many separate components
- The integration of the various components is tested prior to release
- Distribution is being maintained and updated
- As more users use a given distribution the support community around it will steadily increase
- Commercial (paid) support is available for mission critical installations of the most common distributions

The Major Linux Distributions

There are literally hundreds of different Linux distributions available. The available distributions can be divided into two main categories.

Commercial Linux Distributions

- These distributions require a license and commercial support is widely available from various companies.
- The release cycle tends to be much longer and support is available even after a specific version is no longer sold.

Freely Available Distributions

- These distributions are mostly supported by the user community themselves.
- Some have strong ties to commercial products (Fedora, OpenSUSE), which the free distributions are “testing” releases for the commercial versions
- These distributions are more on the “bleeding” edge and incorporate new features more quickly.

Most Linux distributions are created for specific reasons.

- Specialized hardware support, including embedded systems
- Geographical/Cultural needs and requirements
- Project specific

The Major Linux Distributions

Commercial Linux Distributions

- RedHat Linux Enterprise
- Novell SUSE Linux Professional and Enterprise Server
- TurboLinux Desktop and Server
- Mandriva (formerly Mandrake)
- Xandros (it claims to be one of the easiest distributions)
- Linspire (formerly Lindows, tried to bring Linux to the masses, subscription based)

Freely Available Distributions

- Fedora Core (related to RedHat Linux)
- OpenSUSE (related to Novell SUSE)
- CentOS (based of Fedora with stability and security for web hosting in mind)
- Debian (one of the biggest catalog of pre-built applications)
- Ubuntu (a fairly new Debian based distribution, one of the most famous free distributions)
- Gentoo (mostly a source distribution, you have to build it manually)
- Slackware Linux (the oldest distribution)
- Knoppix (famous for the invention of the bootable Linux CD)
- MEPIS (young distribution with many pre-installed applications, can run from CD)

A good online resource to compare and read about many distributions is **www.distrowatch.org**. Currently there are **336** active distributions listed.

Enterprise Grade Linux Distributions

There are a handful enterprise grade distributions available. The two best known ones are

RedHat Enterprise Linux

Novel SUSE Linux Enterprise Server

Some of the advantages of an enterprise level distributions are:

- Good hardware support from PC to mainframe
- Widely available support from various outlets
- Stable, well established, release cycle
- Investment protection by allowing for longer time between upgrades

The TPF organization recommends to use an enterprise grade Linux distribution for your z/TPF build environment, because it is a critical component of your TPF infrastructure.

Linux Distributions - Fedora Core

Website: http://fedora.redhat.com/
Origin: USA
Current Release: Fedora Core 4
Release Date: 2005/06/13
Kernel: 2.6.11

About:

Red Hat created the Fedora project as an openly developed Linux distribution. From the Red Hat perspective the main purpose of the Fedora project is to test new features and functionality and at certain pre-determined time intervals the efforts of this open distribution are rolled into the next release of their Enterprise Linux distribution.

It is a general purpose distribution with excellent community support and wide adoption world wide. It also has very good hardware support. The Fedora project releases about 2 releases per year, which allows it to be fairly up to date at all times.

Linux Distributions - RedHat Enterprise Linux

Website: http://www.redhat.com/software/rhel/
Origin: USA
Current Release: Red Hat Enterprise Linux 4
Release Date: 2005/02/15
Kernel: 2.6.9

About:

RedHat Enterprise Linux is based on the efforts of the Fedora project. It has less frequent releases as the Fedora project though and is focused on stability and maintainability. The longer release cycles are welcomed in enterprise environments to reduce cost and provide a less disruptive environment.

Because of its enterprise audience there is also professional support available from RedHat, for a fee. The longer release cycle also includes support and security fixes for older releases which is appreciated by IT managers who run large number of installations.

Red Hat Enterprise Linux comes in various flavors, tailored for large scale, highly scalable servers, midrange servers and desktop system. Together with SUSE Enterprise Linux it controls the market for mainframe Linux installations and is one of the most successful Linux distributions overall.

Linux Distributions - OpenSuSE

Website: http://www.opensuse.org/
Origin: Germany
Current Release: SUSE Linux 9.3
Release Date: 2005/04/15
Kernel: 2.6.11.4

About:

Due to the great success Red Hat has had with its combination of the open community Fedora project providing a base for its Enterprise Linux offering, SUSE has decided to adopt a similar model.

OpenSUSE is fairly new and provides the latest and greatest of the SUSE Linux Professional distribution. Novell and the community support the OpenSUSE distribution together and provide enhancements, fixes and new features for it. Based on the work being done to OpenSUSE Novell then takes it a step further and releases the commercial SUSE Linux Professional distribution.

Linux Distributions - SUSE Linux Professional

Website: http://www.suse.com/
Origin: Germany
Current Release: SUSE Linux Professional 9.3
Release Date: 2005/04/15
Kernel: 2.6.11.4

About:

SUSE Linux Professional is a desktop oriented distribution, tailored for the enterprise market. However, because it is targeted at the desktop market it has a faster release cycle than the SUSE Linux Enterprise Server distribution and it upgrades packages more quickly.

Based on the huge success that Red Hat has with its Fedora project and the benefits it gains from the large community support, SUSE now has an open community edition of SUSE Linux Professional. After the Professional distribution has been established it graduates to the Enterprise offering.

SUSE and Red Hat share the bulk of the Linux market share and both companies are trying to improve the installation, configuration and management tools. SUSE uses its very good YaST management tool which has a slight advantage over the Red Hat tools because they are better integrated.

Linux Distributions - SUSE Linux Enterprise Server

Website: http://www.suse.com/
Origin: Germany
Current Release: SUSE Linux Enterprise Server 9
Release Date: 2004/08
Kernel: 2.6.5

About:

Comparable with the Red Hat Enterprise Linux distribution the SUSE Linux Enterprise Server distribution is targeted for the enterprise server market. It scales just fine from single CPU departmental servers all the way up to mainframe based, multi CPU, servers.

SUSE has spent quite some effort in supporting mainframe based installations and included good hardware support for those environments. Lately Red Hat has awaken from their sleep and the latest Red Hat Enterprise Linux distribution is getting very close to SUSE.

For quite some time SUSE was mainly dominating the European market while Red Hat dominated the US market. With the purchase of SUSE by Novell this has started to change and both companies are becoming more and more global players.

Linux Distributions - Debian

Website: http://www.debian.org/
Origin: Global
Current Release: Debian 3.1
Release Date: 2005/06/06
Kernel: 2.4.27

About:

Debian focuses on stability and thus has one of the slowest release cycles of all freely available distributions.

It also focuses on providing a very large pre-compiled application library that can be installed very easily through system tools. Because of its focus on stability the provided applications are usually a couple versions backlevel as well. More aggressive repositories exist that provide more recent downloads.

Debian has its own package manager (.deb), which provides a little more flexibility than the RedHat based package manager (.rpm).

Most commercially oriented distributions for the end-user space are based on Debian (KNOPPIX, Linspire, Mepis, Xandros, etc).

Linux Distributions - Ubuntu

Website: http://www.ubuntulinux.org/
Origin: Isle of Man
Current Release: Ubuntu 5.04
Release Date: 2005/04/08
Kernel: 2.6.10

About:

Ubuntu is one of the newest desktop Linux distributions but it has received great community support and has grown faster than any other distribution. It is financially supported by a South African millionaire, which causes some concerns about its survivability if this financial stream should be interrupted.

It is based on the Debian distribution but it provides a far quicker release cycle and even the newest application patches are usually available within hours of their release. This shows the great community support of Ubuntu.

It can be compared to the other Debian derivatives like KNOPPIS, Xandros, Linspire, etc, but the philosophy behind Ubuntu is that an operating system should be free for anyone where those other Debian based distributions usually charge some fee.

Linux Distributions - Gentoo

Website: http://www.gentoo.org/
Origin: USA
Current Release: Gentoo 2005.1
Release Date: 2005/08/08
Kernel: 2.6.12

About:

Gentoo is a very flexible and fast Linux distribution for the desktop but it is one of the most difficult ones to install. Gentoo is mostly a source code distribution which means that most modules and components have to be compiled and installed manually during setup of the system. This however makes it a great learning distribution and the resulting operating system is tailored exactly to the hardware.

It's adoption has been great in the development community but it has not been able to make much inroads otherwise.

Gentoo does not use one of the traditional package managers (rpm or deb) but has it's own, fairly sophisticated package manager which is based on the FreeBSD idea of Ports. Most packages are source code packages although the package manager can handle pre-compiled packages as well.

Linux on the Server

The area where Linux made the biggest inroads are on the server side. Linux distributions contain all the necessary open source components to provide server functionality for a variety of tasks:

HTTP Web Server *	Apache
Windows File Server	Samba
Unix File Server	NFS
Directory Server	OpenLDAP
Database Server	MySQL
Mail Server	Qmail, Exim, Sendmail

- Linux servers cover a range from departmental servers all the way to enterprise wide servers.
- Linux kernel 2.6 allows for much better scalability of servers
- A good portion of the Internet content is being served from Linux based servers
- Google uses a huge server farm to maintain their search database

* There is much more to a Web Server than static HTML pages. Apache works together with a wide variety of add-on components to provide a complete web server infrastructure (JSP, Servlets, PHP, PERL, Python, etc)

L-A-M-P

One of the most frequently used terms in conjunction with Linux servers is

L A M P

L(inux) **A**(pache) **M**(ySQL) **P**(HP, erl, ython)

It abbreviates the frequently used combination of open source software projects to run a web server.

L(inux) The operating system

A(pache) The base HTTP web server that interacts with the client browser or external applications
It is highly customizable, flexible and provides good security. Configuration is based on
text files but there are many open source projects that provide GUI based configuration tools

M(ySQL) A good open source database management tool and server. Customizable and scalable. It is owned and supported by the private company MySQL AB. It provides both a free and a paid distribution and can be licensed under the GPL 2 or other licenses.

P(HP)
P(erl)
P(ython) The letter P stands for a few different scripting languages. Perl used to be the most dominant of them but lately PHP has gained a lot of support from large companies and it is becoming the preferred scripting language these days. Python has its followers but plays more of a niche role. Scripting languages are usually used to create dynamic contents for web sites and because they are interpreted languages they provide a simple development cycle.

Linux on the Desktop

On the desktop side Linux has not had the same impact as on the server side. The dominance of the Windows operating system is hard to penetrate but there are some niche markets where Linux has seen some successes and continues to be a valid alternative to Windows:

- Thin client devices
- Kiosk environments to run custom applications (self service devices)
- Limited functionality environments (airport counters)

A desktop environment these days requires a graphical environment. In Linux, as with other Unix like systems, this is accomplished through the X-Window system with an X-Server. The sometimes confusing part is the fact that the X-Window graphic system that controls the screen is actually a server. The Linux system sends X commands to the server which are translated into graphical operations on the screen.

In most instances the X-Server is located on the same system as the Linux system and the two are looked at as one entity. However, the very nature of the X-Server architecture allows the graphical system to be located on another system altogether, such as a thin client, or even a Windows machine. It's comparable with a Remote Desktop system in Windows only that it has always been part of the X-Server architecture.

Because the client and server are reversed from traditional conceptions, it can cause problems when the Linux system tries to connect to an X-Server through firewalls and NAT devices. This can usually be resolved by tunneling the X traffic over an established SSH sessions (PuTTY) and adds security at the same time.

Linux on the Desktop

An X-Server alone does not provide the type of graphical user environment that we expect. We also need a helper application to provide a useable desktop environment and the two main choices are:

GNOME (GNU Object Model Environment)
Gnome provides an attractive desktop environment with many useful utilities. It also provides Internationalization as well as accessibility features. It also provides a range of development tools that allow anyone to create applications that seamlessly integrate into the Gnome Desktop.

Gnome was created as alternative to KDE because uses the Qt toolkit, which initially was not released under a free software license and thus was not compatible with the rest of the GNU/Linux system. Gnome itself uses the GTK toolkit which is licensed under the LGPL, like Gnome.

KDE
Like Gnome, KDE provides an attractive desktop environment with its own set of utilities. It also has its own set of full blown applications like KOffice which is a Microsoft Office compatible suite. KDE has it's own set of development tools and libraries which allow anyone to develop KDE type applications.

KDE can now be licensed under the GPL.

Most distributions are bundled with both, Gnome and KDE and selection depends on user preference.

Linux for Embedded Systems

Because of it's tremendous portability and customizability, Linux runs on many embedded systems. Linux can be configured to use very little memory and fit on almost any platform.

- Network devices (routers, firewalls, etc)
- Thin clients
- Intelligent sensors
- IP Phones
- Handheld computers

LSB - Linux Standard Base

Problem:

Too many Linux distributions causes support problems because the internal structures are often different

Solution:

LSB - The Linux Standard Base - www.linuxbase.org

- Joint project by several Linux distributions
- Organize internal structures of a Linux system
  - standard libraries
  - file system layout
  - minimum command set
  - minimum set of utilities
  - standardization of runlevels
  - X-Window functionality

- LSB compatible software packages are RPM packages
- Based on the POSIX specification
- Certification done through The Open Group and the Free Standards Group

- Increase compatibility amongst Linux distributions
- Encourage companies to write applications for Linux because of the increased portability

SELinux - Security Enhanced Linux

Problem:

Linux has a fairly simple user and group based access control mechanism

Solution:

SELinux (Security Enhanced Linux) - www.nsa.gov/selinux

- Is a set of definitions, policies and utilities, not a Linux distribution
- Development lead by the US National Security Agency (NSA)
- Open Source project
- Enforces mandatory access controls
- Roles based access control
- Multi level access control
- Tamperproof
- Access controls are stored in extended file attributes (XATTR), which are not supported by all file systems
- Linux kernel 2.6 has full support for SELinux but Linux kernel 2.4 has most functionality through patches
- Red Hat Enterprise Linux 4 currently the only commercial Linux distribution to support it

Linux File System

The Linux file system is slightly different from other operating systems:

- One uniform, hierarchical tree like directory structure
- No drive letters
- Drives are “mounted” into the directory structure

Basic Directory Tree

```
/
├──bin
├──etc
├──home
│   └──dan
├──root
└──var
```

Adding a new hard drive to the system

```
mkdir /home/ztpf
```

```
/
├──bin
├──etc
├──home
│   ├──dan
│   └──ztpf
├──root
└──var
```

```
mount /dev/sdb1 /home/ztpf
```

```
/
├──bin
├──etc
├──home
│   ├──dan
│   └──ztpf
│       └──put01
│           ├──base
│           ├──hpo
│           ├──tpfdf
│           └──tpftools
├──root
└──var
```

Accessing a Linux System

A Linux system can be accessed in different ways:

Telnet

Don't! Telnet is the traditional way of accessing a Linux system command shell but it is highly insecure and should be avoided at all cost

SSH

This is the recommended way to access a command shell on a Linux system. SSH v2 in particular is very secure by encrypting the communication and optionally using certificates to authenticate the Linux system and the client terminal. On Windows the most popular SSH client is PuTTY (www.chiark.greenend.org.uk/~sgtatham/putty)

SSH also allows to tunnel other communication traffic like FTP and X over the same connection. In the case of X it is usually the only solution to pass through network filters, NAT devices and firewalls.

FTP

Don't use FTP directly! Either use the SSH connection to tunnel the FTP traffic or use an integrated solution like WinSCP under Windows.

Accessing a Linux System

WinSCP

This is a great Windows program that provides SFTP (SSH tunneled FTP access). It can exchange authentication information with PuTTY. (www.winscp.net)

X-Server

Using an X-Server directly can work in some instances but security is not always guaranteed. The X connection can be secured with the correct configuration but it is much easier to simply tunnel the traffic over an established SSH session.

VNC

VNC (Virtual Network Computing) is a widely used approach for remote access to computer systems. It is available for a variety of systems, Windows included. Most Linux systems provide a VNC server in their standard configuration and allow a VNC client to access the GUI desktop of the Linux system. Performance is not as good as with a plain X-Server. Some implementations provide encryption but it is not standard.

Locally

Naturally a desktop system is accessed through the local keyboard and screen.

Keeping A Linux System Up To Date

Different distributions provide different mechanisms to keep system components and applications up to date.

Up2date

The Red Hat update utility.

- Checks with a list of configured update servers for updates of installed components
- Enterprise Linux distributions require a subscription to gain full functionality

RedHat Network

This is a premium management offer from Red Hat

- Ability to manage multiple systems
- Remote installation of updated and new applications

Yast

"Yet Another Setup Tool", the SUSE Linux setup and configuration tool

- Provides access to update server to download and install updates
- Ability to load service pack releases

Yum

"Yellow dog Update, Modified" an RPM package manager

- Works with any RPM based Linux distribution (Red Hat, SUSE)
- Updates, installs and removes system components and applications
- A large list of update servers are available on the Internet
- Resolves dependencies
- Can work in conjunction with up2date and yast

Keeping A Linux System Up To Date

Apt

"Advanced Packaging Tool" a Debian package manager

- Similar to Yum but for Debian based systems
- Also available for Red Hat based systems, not very common though
- Updates, installs and removes system components and applications
- A large list of update servers are available on the Internet
- Resolves dependencies

Portage

Gentoo package manager

- Updates, installs and removes system components and applications
- Compiles source packages on the fly (under the covers)
- Used through the "emerge" front end
- Based on the "BSD Ports" idea

CNR

"Click and Run" the Linspire update service

- Updates, installs and removes system components and applications
- Requires subscription

z/TPF And Linux

As outlined several times by various other presentations, z/TPF utilizes a Linux system for:

Building
- Linux utility programs
- SIP deck assembly
- Face table generation
- Assemble, compile, link of all online programs
- Assemble, compile, link of all Linux offline programs

Loading
- OLDR loads (via FTP) to z/TPF

Access to source code
- SCM integration
- Serve source code for z/OS and Windows
- Editing of source code

Which Distribution And Where To Run It

Which Distribution

- As stated earlier the TPF organization recommends an Enterprise level Linux distribution.
- While there are a handful of options available we recommend

RedHat Enterprise Linux

or

Novell SUSE Linux Enterprise Server

Where to run it

Ideally you should be able to run your z/TPF Linux build environment on any hardware platform….

…but:

- For assembly tasks IBM only has HLASM available for zSeries systems, however, 3rd party alternatives exist for other platforms.
- Because of differences in endianism (the order in which multibyte numbers are stored) some of the provided tools only operate correctly on zSeries systems at this time. (any other big endian based system should work in theory but this has not been validated and tested).

Which Distribution And Where To Run It

Recommendation:

RedHat Enterprise Linux v4 on zSeries (or the most current at time of installation)

or

Novell SUSE Linux Enterprise Server v9 on zSeries (or the most current at time of installation)

C/C++ Program Development:

- GNUPro GCC Cross Compiler can be built to run on PC based Linux systems
- MakeTPF build scripts run on PC based Linux systems
- Offldr will be updated in one of the next PUT releases to run on PC based Linux systems

Cross Compilation for z/TPF

- TPF has always been a runtime system only
- Minor differences exist between the z/TPF runtime environment and a s390x runtime environment, which required its own compilation target (s390x-ibm-tpf).
- Red Hat GNUPro 3.4 GCC compiler incorporates all required modification and fixes
- Shortly GCC 4 will contain all the z/TPF specific modifications as well (most likely GCC 4.1)

Linux Based Tools z/TPF Provides

The following are the main tools used to build z/TPF programs

maketpf

maketpf is the command used to assemble, compile, and link TPF programs, including on-line applications, CIMR components, keypoints, and off-line utilities. maketpf is based on schema that utilizes the GNU implementation of make, customized to support the TPF source code hfs structure and define a fixed set of assemble, compile, and link rules.

bldtpf

The bldtpf command is used to generate the z/TPF system configuration data (via SIP and FCTB), build the z/TPF online, offline, and utility programs, generate STUB and IPAT entries, and perform pat-to-control file conversions. These actions are driven using either the control files or the SIP decks as the primary input.

loadtpf

The loadtpf command is used to generate an OLDR load file using the offldr program and ftp it to a TPF system. A load deck or load file can either be specified as input or can be generated from a control file, program name, or shared object name provided as input. This command is supported on Linux only. Also, the userid and password needed to log in to the specified IP address must be specified in a .netrc file, as discussed during the previous post installation setup section.

Linux Based Tools z/TPF Provides

These are additional tools, some are called under the covers when maketpf/bldtpf/loadtpf commands are executed

bpatch_IBM2047 / bpatch	corrects the defective IBM1047 codepage and names it IBM2047
calst	codepage translator for assembler listing files
elfaggr	Compare data structure offsets between C/C++ and assembler DSECTs
fctbg	Face Table generator, generates goff file
glinit	Format 2 global initialization tool
goff2elf	Converts goff object files into elf object files
mqtrace	MQ trace data post processor
offldr	creates OLDR files to load programs, loadtpf uses this
pat2ctl	create bldtpf control file from IPAT
ppsadump	dump post processor
pptrlg	ECB trace log post processor
salo	system allocator program creates program attribute table (PAT)
tpf-localdef	creates locale source files
tpfdb2ppl	DB2 precompiler / post processor
tpfobjpp	elf object post processor, used to insert comments
tpfsochk	checks shared object library files for objects that need relocation
tpfsoexp	create LD export file from shared object files
tpfsym	add debug information into assembler program object files

How Much Linux Knowledge Should I Have

There is not an easy answer to this question but here a suggestion:

Regular Programmer/Developer

- If TPF Toolkit is being used then the Linux knowledge is not really important.
- Programmer does not even have to log on to the system through a command shell
- If using Linux command line tools to edit and built applications some basic Linux knowledge is required. At least enough to navigate the system and to manipulate files

Team Leader / Release Manager

- Should have basic Linux knowledge to navigate the system
- Should know about file/directory permissions

Build System Administrator

- Should have a good understanding about Linux, feel comfortable on the command line
- User and group management
- File/directory permissions and manipulation
- Shell customization, shell scripting
- GCC and make knowledge and practical experience

Where To Go For Help

Linux Distributor

For enterprise level Linux distributions you probably already have a support agreement in place

Linux Distribution Website

Each Linux distribution provides detailed information and links to other support sites

Internet Search Engine

A good search engine will become your best friend

IBM

We provide a wide variety of Linux support

www.distrowatch.org

Good repository of information and news about almost any Linux distribution and links to relevant support and community Websites for a particular distribution

www.gnu.org

The main page to all the GNU tools

www.kernel.org

The home of the Linux kernel

Books

Plenty of choices

Q & A

Any Questions?

Trademarks

IBM, zSeries, S/390, and Rational are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel Inside (logos), MMX, Celeron, Intel Centrino, Intel Xeon, Itanium, Pentium and Pentium III Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Notes
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.