INTRODUCTION TO LINUX

Developed for the Azera Group

By: Joseph D. Fournier B.Sc.E.E., M.Sc.E.E.

Some History: MULTICS-One

- **Multics** ("**Multiplexed Information and Computing Service**") is one of the most influential early time-sharing operating system based on the concept of a single-level memory. It has been said that Multics "has influenced all modern operating systems since, from microcomputers to mainframes."
- Initial planning and development for Multics started in 1964, in Cambridge, Massachusetts. Originally it was a cooperative project led by MIT (Project MAC) along with General Electric and Bell Labs. It was developed on the GE 645 computer, which was specially designed for it; the first one was delivered to MIT in January, 1967.
- Multics was conceived as a commercial product for General Electric, and became one for Honeywell, albeit not very successfully. Due to its many novel and valuable ideas, Multics has had a significant influence on computer science despite its perceived faults.

Some History: MULTICS-Two

- Multics has numerous features intended to ensure high availability so that it would support a computing utility similar to the telephone and electricity utilities.
- Modular hardware structure and software architecture are used to achieve this.
- The system can grow in size by simply adding more of the appropriate resource, be it computing power, main memory, or disk storage.
- Separate access control lists on every file provide flexible information sharing, but complete privacy when needed.
- Multics has a number of standard mechanisms to allow engineers to analyze the performance of the system, as well as a number of adaptive performance optimization mechanisms.

Some History: MULTICS-Three

- Multics has a number of standard mechanisms to allow engineers to analyze the performance of the system, as well as a number of adaptive performance optimization mechanisms.
- Bell Labs pulled out of the project in 1969; some of the people who had worked on it there went on to create the Unix system. Multics development continued at MIT and General Electric. At MIT in 1975, use of Multics was declining and did not recover by 1976 to prior levels.[15][16] Finally by slashing prices, MIT managed to lure users back to Multics in 1978.[17]
- The design and features of Multics were highly influential in the design of the Unix operating system. Unix was originally written by two Multics programmers, Ken Thompson and Dennis Ritchie.
- The influence of Multics on the early Unix Operating System is evident in many areas, including the naming of some commands.

Some History: UNIX-One

- The Unix operating system was conceived and implemented in 1969, at AT&T's Bell Labs, in the United States by Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna. Unix was originally written by two Multics programmers, Ken Thompson and Dennis Ritchie.
- First released in 1971, Unix was written entirely in assembly language. In 1973 in a key, pioneering approach, it was rewritten in the C programming language by Dennis Ritchie (with the exception of some hardware and I/O routines). The availability of a high-level language implementation of Unix made its porting to different computer platforms easier.
- Due to an earlier antitrust case forbidding it from entering the computer business, AT&T was required to license the operating system's source code to anyone who asked. As a result, Unix grew quickly and became widely adopted by academic institutions and businesses.
- In 1984, AT&T divested itself of Bell Labs; freed of the legal obligation requiring free licensing, Bell Labs began selling Unix as a proprietary product, where users were not legally allowed to modify Unix.

Some History: BSD-One

- The earliest distributions of Unix from Bell Labs in the 1970s included the source code to the operating system, allowing researchers at universities to modify and extend Unix. The operating system arrived at Berkeley in 1974.
- A PDP-11/45 (Designed and sold by **Digital Equipment Corporation**) was bought to run the system, but for budgetary reasons, this machine was shared with the mathematics and statistics groups at Berkeley, who used RSTS, (a multi-user time-sharing operating system), so that Unix only ran on the machine eight hours per day
- A larger PDP-11/70 was installed at Berkeley the following year, using money from the Ingres database project.
- Understanding BSD (Berkeley Software Distribution) requires delving far back into the history of Unix.
- BSD began life as a variant of Unix that programmers at the University of California at Berkeley, initially led by Bill Joy, began developing in the late 1970s.

Some History: BSD-Two

- At first, BSD was not a clone of Unix, or even a substantially different version of it. It just included some extra features, which were intertwined with code owned by AT&T.
- In 1975, Ken Thompson took a sabbatical from Bell Labs and came to Berkeley as a visiting professor. He helped to install Version 6 Unix and started working on a Pascal implementation for the system.
- Graduate students Chuck Haley and Bill Joy improved Thompson's Pascal and implemented an improved text editor. Other universities became interested in the software at Berkeley.
- In 1977 Joy started compiling the first Berkeley Software Distribution (1BSD), which was released on March 9, 1978. 1BSD was an add-on to Version 6 Unix rather than a complete operating system in its own right.
- The second Berkeley Software Distribution (2BSD), released in May 1979, included updated versions of the 1BSD software as well as two new programs by Joy that persist on Unix systems to this day: the vi text editor and the C shell.

Some History: BSD-Three

- Several operating systems are based on BSD, including FreeBSD, OpenBSD, NetBSD, MidnightBSD, GhostBSD, Darwin and DragonFly BSD. Both NetBSD and FreeBSD were created in 1993. They were initially derived from 386BSD (also known as "Jolix"), and merged the 4.4BSD-Lite source code in 1994. OpenBSD was forked from NetBSD in 1995, and DragonFly BSD was forked from FreeBSD in 2003.
- BSD was also used as the basis for several proprietary versions of Unix, such as Sun's SunOS, Sequent's DYNIX, NeXT's NeXTSTEP, DEC's Ultrix and OSF/1 AXP (now Tru64 UNIX).
- NeXTSTEP later became the foundation for Apple Inc.'s macOS.

Some History: MINIX-One

- Andrew S. Tanenbaum created MINIX at Vrije Universiteit in Amsterdam to exemplify the principles conveyed in his textbook, *Operating Systems: Design and Implementation* (1987).
- An abridged 12,000 lines of the C source code of the kernel, memory manager, and file system of MINIX 1.0 are printed in the book. Prentice-Hall also released MINIX source code and binaries on floppy disk with a reference manual. MINIX 1 was system-call compatible with Seventh Edition Unix.
- Tanenbaum originally developed MINIX for compatibility with the IBM PC and IBM PC/AT 8088 microcomputers available at the time.
- Minix 3 was publicly announced on 24 October 2005 by Tanenbaum during his keynote speech at the Association for Computing Machinery (ACM) Symposium on Operating Systems Principles (SOSP). Although it still serves as an example for the new edition of Tanenbaum's textbook, coauthored by Albert S. Woodhull, it is comprehensively redesigned to be "usable as a serious system on resource-limited and embedded computers and for applications requiring high reliability."

Some History: MINIX-Two

- Minix 3 currently supports IA-32 and ARM architecture systems. It is available in a live CD format that allows it to be used on a computer without installing it on the hard drive, and in versions compatible with hardware emulating and virtualizing systems, including Bochs, QEMU, VMware Workstation and Fusion, VirtualBox, and Microsoft Virtual PC.
- Version 3.1.2 was released on 18 April 2006. It was the first version after MINIX had been relicensed under the BSD-3-Clause license.
- Version 3.1.5 was released on 5 November 2009. It contains X11, emacs, vi, cc, gcc, perl, python, ash, bash, zsh, ftp, ssh, telnet, pine, and over 400 other common Unix utility programs.
- With the addition of X11, this version marks the transition away from a text-only system. In many cases it can automatically restart a crashed driver without affecting running processes.

Some History: MINIX-Three

- MINIX is self-healing and can be used in applications demanding high reliability. MINIX 3 also has support for virtual memory management, making it suitable for desktop OS use.
- As of version 3.2.0, the userland was mostly replaced by that of NetBSD and support from pkgsrc became possible, increasing the available software applications that MINIX can use.
- Clang replaced the prior compiler (with GCC now having to be manually compiled), and GDB, the GNU debugger, was ported.
- Minix 3.3.0, released in September 2014, brought ARM support.
- Minix 3.4.0RC, Release Candidates became available in January 2016;
- Minix supports many programming languages, including C, C++, FORTRAN, Modula-2, Pascal, Perl, Python, and Tcl.
- Minix 3 still has an active development community with over 50 people attending MINIXCon 2016, a conference to discuss the history and future of MINIX.
- All Intel chipsets post-2015 are running MINIX 3 internally as the software component of the Intel Management Engine.

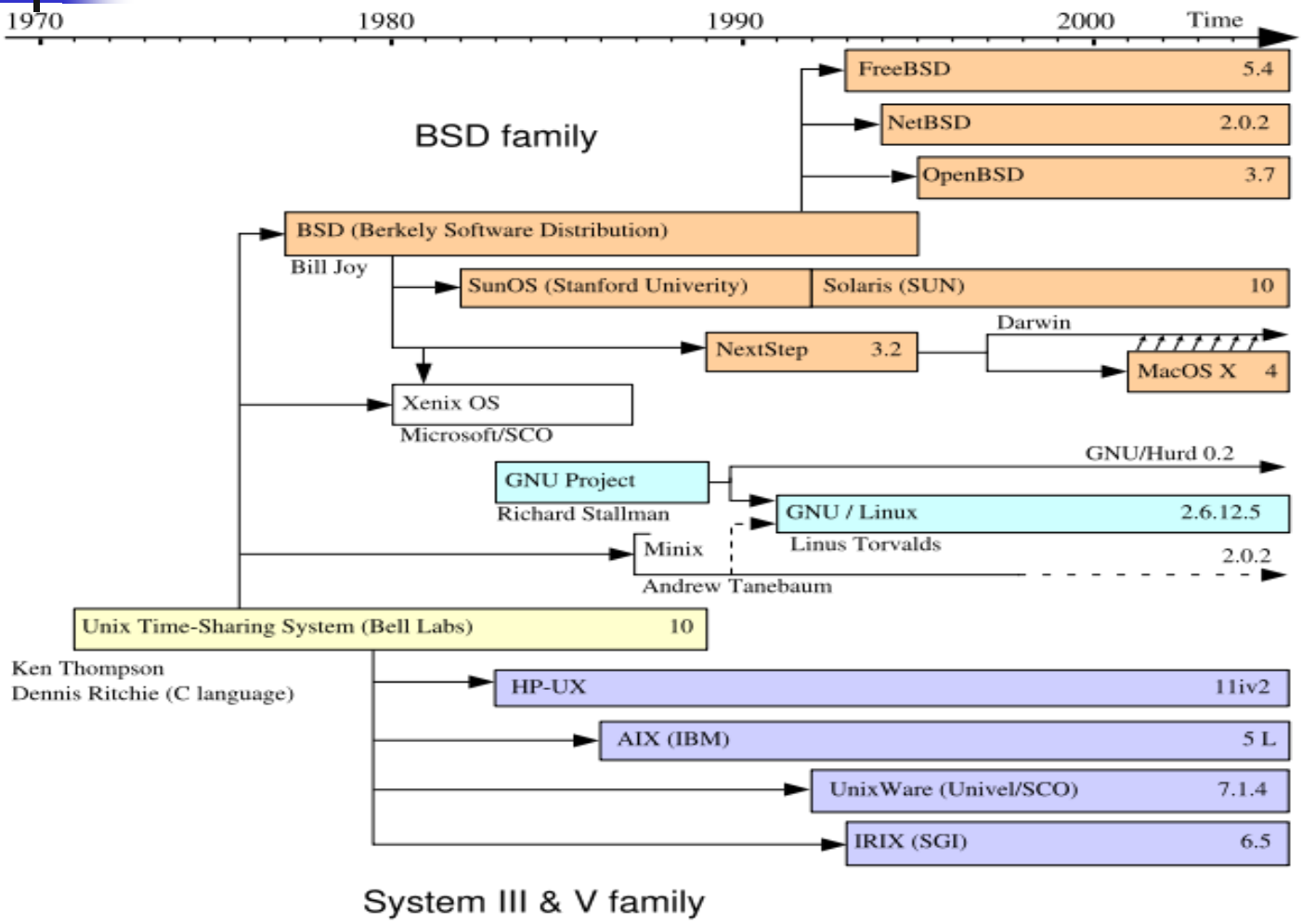
Minix and Linux

- Linus Torvalds used and appreciated Minix, but his design deviated from the Minix architecture in significant ways, most notably by employing a monolithic kernel instead of a microkernel. This was disapproved of by Tanenbaum in the Tanenbaum–Torvalds debate. Tanenbaum explained again his rationale for using a microkernel in May 2006.
- Early Linux kernel development was done on a Minix host system, which led to Linux inheriting various features from Minix, such as the Minix file system.
- Linux was originally developed for personal computers based on the Intel x86 architecture, but has since been ported to more platforms than any other operating system.[22] Because of the dominance of the Linux-based Android on smartphones, Linux also has the largest installed base of all general-purpose operating systems.

UNIX

- **Unix** (officially trademarked as **UNIX**®) is a computer operating system originally developed in the 1960s and 1970s by a group of AT&T employees at Bell Labs including Ken Thompson, Dennis Ritchie and Douglas McIlroy. Today's Unix systems are split into various branches, developed over time by AT&T as well as various commercial vendors and non-profit organizations.
- The present owner of the trademark *UNIX*® is The Open Group, an industry standards consortium. Only systems fully compliant with and certified to the Single UNIX Specification qualify as "UNIX®" (others are called "Unix system-like" or "Unix-like").
- During the late 1970s and early 1980s, Unix's influence in academic circles led to large-scale adoption (particularly of the BSD variant of Unix) by commercial startups, the most notable of which is Sun Microsystems. Today, in addition to certified Unix systems, Unix-like operating systems such as Linux, Mac OS X and BSD derivatives are commonly encountered.

UNIX


1970
1980
1990
2000
Time
FreeBSD 5.4
NetBSD 2.0.2
OpenBSD 3.7
BSD family
BSD (Berkely Software Distribution)
Bill Joy
SunOS (Stanford Univerity)
Solaris (SUN) 10
Darwin
NextStep 3.2
MacOS X 4
Xenix OS
Microsoft/SCO
GNU/Hurd 0.2
GNU Project
Richard Stallman
GNU / Linux 2.6.12.5
Linus Torvalds
Minix
2.0.2
Andrew Tanebaum
Unix Time-Sharing System (Bell Labs) 10
Ken Thompson
Dennis Ritchie (C language)
HP-UX 11iv2
AIX (IBM) 5 L
UnixWare (Univel/SCO) 7.1.4
IRIX (SGI) 6.5
System III & V family

Unix Shell

- A **Unix shell**, also called "the command line", provides the traditional user interface for the Unix operating system and for Unix-like systems.
- bash – Bourne Again SHell, (mostly) sh-compatible and csh-compatible, standard shell on Linux systems and Mac OS X.
- csh – C shell. Written by Bill Joy for BSD systems.
- ksh – Korn shell, standard shell on many proprietary Unix systems, powerful successor to the Unix Bourne shell (sh), written by David Korn,
- rc – originally written for Plan 9.
- sh – Bourne shell, only shell present on all UNIX and Unix-like systems, written by Stephen Bourne for Version 7 Unix.
- tcsh – TENEX C shell, standard shell on BSD systems.
- zsh – Z shell.

GNU-One

- The GNU project was publicly announced by Richard Stallman IN September 1983. Stallman announced the plan for the GNU operating system on several ARPANET mailing lists and USENET.
- He started the project on his own and describes: "As an operating system developer, I had the right skills for this job. So even though I could not take success for granted, I realized that I was elected to do the job. I chose to make the system compatible with Unix so that it would be portable, and so that Unix users could easily switch to it."
- In 1985, Stallman published the GNU Manifesto, which outlined his motivation for creating a free operating system called GNU, which would be compatible with Unix. The name GNU is a recursive acronym for "GNU's Not Unix".

GNU-Two

- The same year he started a nonprofit corporation called the Free Software Foundation to employ free software programmers and provide a legal infrastructure for the free software movement. Stallman was the non-salaried president of the FSF, which is a 501(c)(3) nonprofit organization founded in Massachusetts.
- GNU was to be a complete Unix-like operating system composed entirely of free software. Software development work began in January 1984. By the beginning of the 1990s, the project had produced or collected most of the necessary components of this system, including libraries, compilers, text editors, and a Unix shell.
- Thus the GNU mid-level portions of the operating system were almost complete. The upper level could be supplied by the X Window System, but the lower level, which consisted of a kernel, device drivers, and daemons, was still mostly lacking. In 1990, the GNU project began developing the GNU Hurd kernel, based on the Mach microkernel.

Linux-Introduction

- In 1991, work on the Linux kernel began by Linus Torvalds while he was attending the University of Helsinki.
- Torvalds originally created the Linux kernel as a non-commercial replacement for the Minix kernel; he later changed his original non-free license to the GPLv2, which differed primarily in that it also allowed for commercial redistribution.
- Although dependent on the Minix userspace at first, work from both Linux kernel developers and the GNU project allowed Linux to work with GNU components. Thus Linux filled the last major gap in running a complete, fully functional operating system built from free software.

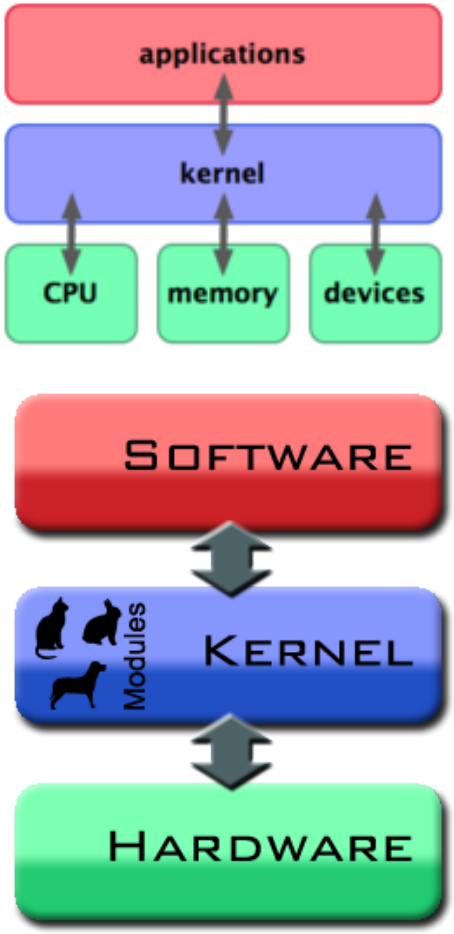
Linus Torvalds

The Linux Kernel

- Linux was originally developed for personal computers based on the Intel x86 architecture, but has since been ported to more platforms than any other operating system.
- Because of the dominance of the Linux-based Android on smartphones, Linux also has the largest installed base of all general-purpose operating systems.
- Although Linux is used by only around 2.3 percent of desktop computers, the Chromebook, which runs the Linux kernel-based Chrome OS, dominates the US K–12 education market and represents nearly 20 percent of sub-$300 notebook sales in the US.
- Linux is the leading operating system on servers (over 96.4% of the top 1 million web servers' operating systems are Linux), leads other big iron systems such as mainframe computers, and is the only OS used on TOP500 supercomputers (since November 2017, having gradually eliminated all competitors).

Kernel-Introduction

- In computing, the **kernel** is the central component of most computer operating systems (OSs). Its responsibilities include managing the system's resources and the communication between hardware and software components.
- As a basic component of an operating system, a kernel provides the lowest-level abstraction layer for the resources (especially memory, processors and I/O devices) that applications must control to perform their function. It typically makes these facilities available to application processes through inter-process communication mechanisms and system calls.
- While monolithic kernels will try to achieve these goals by executing all the code in the same address space to increase the performance of the system, microkernels run most of their services in user space, aiming to improve maintainability and modularity of the codebase.



Linux Distributions

- A **Linux distribution**, often simply **distribution** or **distro**, is a member of the Linux family of Unix-like operating systems comprised of the Linux kernel, the non-kernel parts of the GN operating system, and assorted other software.
- Linux distributions take a variety of forms, from fully-featured desktop and server operating systems to minimal environments (typically for use in embedded systems, or for booting from a floppy).
- The most common Linux Distributions are: CentOS, Debian, Fedora, Gentoo, Knoppix, Mandriva Linux, Red Hat Enterprise Linux (Now owned by IBM), Slackware (One of the oldest distributions still maintained and used), SUSE Linux and Ubuntu.

Bash-The Linux Shell

- **Bash** is a Unix shell written for the GNU Project. The name of the actual executable is **bash**.
- Its name is an acronym for *Bourne-again shell.* The Bourne shell (sh), was an early and important Unix shell written by Stephen Bourne and distributed with Version 7 Unix circa 1978. Bash was created in 1987 by Brian Fox, using the Bourne shell as a starting point of design.
- Bash is the default shell on most Linux systems as well as on Mac OS X and it can be run on most Unix-like operating systems.
- It has also been ported to Microsoft Windows within the Cygwin POSIX emulation environment for Windows.
- Released under the GNU General Public License, Bash is free software.

Linux Filesystem-One

- The Linux operating systems creates a virtual file system, which makes all the files on all the devices appear to exist in a single hierarchy. This means that there is one root directory, and every file existing on the system is located under it somewhere.
- The Linux operating system assign a device name to each device, but this is not how the files on that device are accessed. Instead, to gain access to files on another device, the operating system must first be informed where in the directory tree those files should appear.
- This process is called mounting a file system. For example, to access the files on a CD-ROM, one must tell the operating system "Take the file system from this CD-ROM and make it appear under such-and-such directory." The directory given to the operating system is called the *mount point* . It may be empty, or it may contain subdirectories for mounting individual devices.
- Generally, only the administrator (i.e. root user) may authorize the mounting of file systems.
- The Linux operating system includes software and tools that assist in the mounting process and provide it new functionality.

Linux Filesystem-Two

- In many situations, file systems other than the root need to be available as soon as the operating system has booted. The system therefore provide a facility for mounting all user file systems at boot time.
- Removable media allow programs and data to be transferred between machines without a physical connection.
- **Supermounting**; For example, a USB drive that has been supermounted can be physically removed from the system. Under normal circumstances, the disk should have been synchronized and then unmounted before its removal. Provided synchronization has occurred, a different disk can be inserted into the drive. The system automatically notices that the disk has changed and updates the mount point contents to reflect the new medium.
- The automounter will automatically mount a file system when a reference is made to the directory atop which it should be mounted. This is usually used for file systems on network servers, rather than relying on events such as the insertion of media, as would be appropriate for removable media.

Filesystem Utilities-One

- info – The GNU alternative to man
- man – The standard unix documentation system
- chattr – Change file attributes on a Linux second extended file system
- chgrp – Change the group of a file or directory
- chmod – Change the permissions of a file or directory
- chown – Change the owner of a file or directory
- cd – Change to another directory location
- cp – Copy a file or directory to another location

Filesystem Utilities-Two

- df – Report disk space
- dircmp – Compare contents of files between two directories
- du – Calculate used disk space
- fdupes – Find or remove duplicate files within a directory
- find – Search for files through a directory hierarchy
- fsck – Filesystem check
- ln – Link one file/directory to another
- ls – List directory contents
- lsattr – List file attributes on a Linux second extended file system
- lsof – list open files

Filesystem Utilities-Three

- mkdir – Make a directory
- mkfifo – Make a named pipe
- mount – Mount a filesystem
- mv – Move or rename a file or directory
- pwd – Print the current working directory
- rm – Delete a file or directory tree
- readlink – Display value of a symbolic link, or display canonical path for a file
- rmdir – Delete an empty directory
- touch – Create a new file or update its modification time
- tree – Print a depth-indented tree of a given directory
- unlink – System call to remove a file or directory

Text Editors

- GNU Emacs – Freely programmable full-screen text editor and general computing environment (using built-in Elisp, a simple dialect of the Lisp programming language)
- Joe – a screen-oriented text editor using a Wordstar-style command set
- Jove – a screen-oriented text editor using an Emacs-style command set
- pico – PIne's message COmposition editor (simple, easy to use screen editor)
- vi – "Visual" (screen-oriented) text editor (originally ex in screen-oriented "visual" mode)
- vim – Vi IMproved, portable vi-compatible editor with multiple buffers, screen splitting, syntax highlighting and a lot of other features not found in standard ex/vi
- XEmacs – Popular version of emacs that is derived from GNU Emacs

Text Processing-One

- awk – A pattern scanning and processing language
- banner – Creates ascii art version of an input string for printing large banners
- cat – Concatenate files to standard output
- cksum – Print the CRC checksum and bytecount of a file (see also MD5)
- cmp – Compare two files byte for byte
- cut – Remove sections from each line of a file or standard input
- diff – Compare two text files line by line
- egrep – Extended pattern matching (synonym for "grep -E")
- fgrep – Simplified pattern matching (synonym for "grep -F")
- fold – Wrap each input line to fit within the given width
- grep – Print lines matching a pattern
- head – Output the first parts of a file
- iconv – Convert the encoding of the specified files
- join – Join lines of two files on a common field
- less – Improved more-like text pager

Text Processing-Two

- more – Pager
- nroff – Fixed-width (non-typesetter) version of the standard Unix typesetting system
- patch – Change files based on a patch file
- sed – Stream EDitor
- sort – Sort lines of text files
- split – Split a file into pieces
- tail – Output the tail end of files
- tee – Read from standard input, write to standard output and files
- uudecode – Decodes a binary file that was used for transmission using electronic mail
- uuencode – Encodes a binary file for transmission using electronic mail
- wc – Word/line/byte count

Archivers and compression

- afio – Compatible superset of cpio with added functionality
- ar – Maintain, modify, and extract from archives. Now largely obsoleted by tar
- bzip2 – Block-sorting file compressor
- compress – Traditional compressor using the LZW algorithm
- cpio – A traditional archiving tool/format
- gzip – The gzip file compressor
- p7zip – 7zip for unix/linux
- pack, pcat, unpack – included in old versions of ATT Unix. Uses Huffman coding, obsoleted by compress.
- pax – POSIX archive tool that handles multiple formats.
- tar – Tape ARchiver, concatenates files
- uncompress – Uncompresses files compressed with *compress*.
- zcat – Prints files to stdout from gzip archives without unpacking them to separate file(s)

Communication

- ftp, sftp – File transfer protocol, secure FTP
- NFS – Network filesystem
- OpenVPN – virtual private (encrypting) networking software
- Postfix — mail transfer agent
- rsh, SSH, telnet – Remote login
- Samba – SMB and CIFS client and server for UNIX
- Sendmail – popular E-Mail transport software
- talk – Talk to another logged-in user
- uustat – a Basic Networking Utilities (BNU) command that displays status information about several types of BNU operations
- uux – Remote command execution over UUCP

Network monitoring and security

- John the Ripper – password cracking software
- Nessus – a comprehensive open-source network vulnerability scanning program
- Netstat – displays a list of the active network connections the computer
- Nmap – free port scanning software
- SAINT – System Administrator's Integrated Network Tool – Network Vulnerability Scanner.
- SATAN – the Security Administrator Tool for Analyzing Networks – a testing and reporting tool that collects information about networked hosts
- Snort – an open source network intrusion detection system
- tcpdump – a computer network debugging tool that intercepts and displays TCP/IP packets being transmitted or received
- Wireshark – a protocol analyzer, or "packet sniffer", similar to tcpdump, that adds a GUI frontend, and more sorting and filtering options.

Remote Login: SSH Client

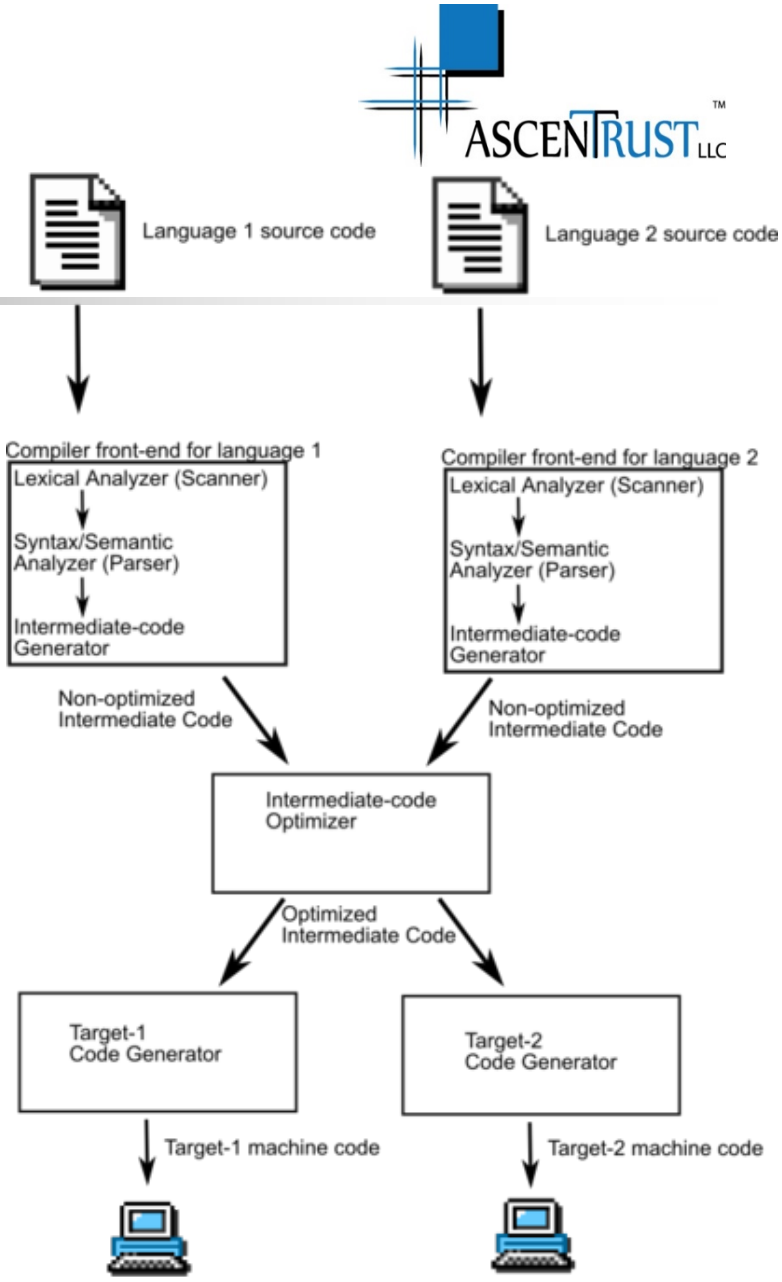
- SSH applications are based on a client–server architecture, connecting an SSH client instance with an SSH server.
- SSH operates as a layered protocol suite comprising three principal hierarchical components:
  - the *transport layer* provides server authentication, confidentiality, and integrity;
  - the *user authentication protocol* validates the user to the server;
  - the *connection protocol* multiplexes the encrypted tunnel into multiple logical communication channels.
- SSH was designed on Unix-like operating systems, as a replacement for Telnet and for unsecured remote Unix shell protocols, such as the Berkeley Remote Shell (rsh) and the related rlogin and rexec protocols, which all use insecure, plaintext transmission of authentication tokens.

Programming Tools

- bash – Bourne Again SHell, (mostly) sh-compatible and csh-compatible, standard shell on Linux systems and Mac OS X.
- csh – C shell. Written by Bill Joy for BSD systems.
- ksh – Korn shell, standard shell on many proprietary Unix systems, powerful successor to the Unix Bourne shell (sh), written by David Korn,
- rc – originally written for Plan 9.
- sh – Bourne shell, only shell present on all UNIX and Unix-like systems, written by Stephen Bourne for Version 7 Unix.
- tcsh – TENEX C shell, standard shell on BSD systems.
- zsh – Z shell.
- awk – Standard Unix pattern scanning and text processing tool.
- perl – Perl scripting language.
- PHP – PHP scripting language.
- Python – Python scripting language.

Compiler-One

- A **programming language** is an artificial language that can be used to control the behavior of a machine, particularly a computer.
- A **compiler** is a computer program (or set of programs) that translates text written in a computer language (the *source language*) into another computer language (the *target language*).
- The original sequence is usually called the *source code* and the output called *object code*.
- Commonly the output has a form suitable for processing by other programs (e.g., a linker), but it may be a human-readable text file.



Compiler-Two

- as – GNU assembler tool.
- c99 – C programming language.
- cc – C compiler.
- dbx – (System V and BSD) Symbolic debugger.
- f77 – Fortran 77 compiler.
- gcc – GNU Compiler Collection C frontend (also known as GNU C Compiler)
- gdb – GNU symbolic debugger.
- ld – Program linker.
- lex – Lexical scanner generator.
- ltrace – (Linux) Trace dynamic library calls in the address space of the watched process.
- m4 – Macro language.
- make – Automate builds.
- nm – List symbols from object files.
- size – return the size of the sections of an ELF file.
- strace – (Linux) or truss (Solaris) Trace system calls with their arguments and signals. Useful debugging tool, but does not trace calls outside the kernel, in the address space of the process(es) being watched.

Desktops/Graphical User Interfaces

The term "user interface" is often used in the context of (personal) computer systems and electronic devices. Where a network of equipment or computers are interlinked through an MES (Manufacturing Execution System)-or Host to display information.

- A human–machine interface (HMI) is typically local to one machine or piece of equipment, and is the interface method between the human and the equipment/machine.
- An operator interface is the interface method by which multiple pieces of equipment that are linked by a host control system are accessed or controlled.
- The system may expose several user interfaces to serve different kinds of users. For example, a computerized library database might provide two user interfaces, one for library patrons (limited set of functions, optimized for ease of use) and the other for library personnel (wide set of functions, optimized for efficiency).

Desktops/Graphical User Interfaces

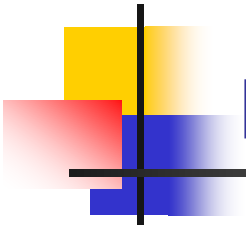
- CDE – Common Desktop Environment, most commonly found on proprietary UNIX systems
- Enlightenment – an open source window manager for the X Window System
- FVWM and its variant.
- GNOME – GNU Network Object Model Environment
- IceWM – ICE Window Manager
- JWM – Joe's Window Manager
- KDE – K Desktop Environment
- XFce – a desktop environment for Unix and other Unix-like platforms

Package Management

- apt – Front-end for dpkg or rpm
- debconf – Debian package configuration management system
- dpkg – The Debian package manager
- drakconf – Front-end configuration utility for Mandriva Linu
- emerge – A frontend to portage
- pacman – A package manager used primarily by Arch Linux
- portage – The Gentoo Linux package manager
- rpm – Originally the package manager for Red Hat Linux, now used by several distributions including Mandriva Linux
- Synaptic – GTK+ frontend for the apt package manager. Primarily used by Ubuntu Linux, Debian Sarge, and other Debian-based systems; but usable on any system using apt.
- urpmi – Front-end to rpm, used by Mandriva Linux
- YaST - System management utility mainly used by SuSE
- yum - Front-end for rpm, used by Fedora and CentOS

Web Browsers

- Dillo – Extremely light-weight web browser
- ELinks – Enhanced links
- Epiphany – Light-weight GNOME web browser
- Galeon – Light-weight old GNOME web browser
- Konqueror – KDE web browser
- Links – Console based web browser
- lynx – Console based web browser
- Mozilla Application Suite – Graphical cross platform web browser & email client
- Mozilla Firefox – Extensible Web browser
- Opera – Web browser and e-mail client
- w3m – Console based web browser



Desktop Publishing

- groff – Traditional typesetting system
- LaTeX – Popular TeX macro package for higher-level typesetting
- lp – Print a file (on a line printer)
- Passepartout – Desktop publishing program
- pr – Convert text files for printing
- Scribus – Desktop publishing program
- TeX – Macro-based typesetting system
- troff – The original and standard Unix typesetting system

Math Tools

- maxima – Symbol manipulation program.
- Octave – Numerical computing language (mostly compatible with Matlab) and environment.
- R – Statistical programming language.
- units – Unit conversion program.
- bc – An arbitrary precision calculator language with syntax similar to the C programming language.
- cal – Displays a calendar
- dc – Reverse-Polish desk calculator which supports unlimited precision arithmetic
- fortune – Fortune cookie program that prints a random quote

Kernel Specific Commands

Kernel specific

- date – Print or set the system date and/or time
- dmesg – Print the kernel message buffer
- ipcrm – Remove a message queue, semaphore set or shared memory id
- ipcs – Provide information on IPC facilities
- uname – Print assorted system statistics

General User Commands

- dd – Convert and copy a file (Disk Dump)
- dirname – Strip non-directory suffixes from a path
- echo – Print to standard output
- env – Show environment variables; run a program with altered environment variables
- file (or stat) – Determine the type of a file
- nohup – Run a command with immunity to hangups outputting to non-tty
- sh – The Bourne shell, the standard Unix shell
- uptime – Print how long the system has been running

Processes and tasks management

- anacron – Periodic command scheduler
- at – Single-time command scheduler
- chroot – Change the system root directory for all child processes
- cron – Periodic command scheduler
- crontab – Crontab file editor
- daemonic – Interface to daemon init scripts
- htop – Interactive ncurses-based process viewer that allows scrolling to see all processes and their full command lines
- kill – Send a signal to process, or terminate a process (by PID)
- killall – Terminate all processes (in GNU/Linux, it's kill by name)

Processes and tasks management

- nice – Alter priorities for processes
- pgrep – Find PIDs of processes by name
- pidof – GNU/Linux equivalent of pgrep
- pkill – Send a signal to process, or terminate a process (by name). Equivalent to Linux killall
- ps – Report process status
- renice – Alter the priorities of an already running process
- sleep – Delay for specified time
- time – Time a command
- timex – Time process shell execution, measure process data and system activity
- top – Produce a dynamic list of all resident processes
- wait – Wait for the specified process

User management and support

- chsh – Change user shell
- finger – Get details about user
- id – Print real/effective UIDs/GIDs
- last – show listing of last logged in users
- lastlog – show last log in information for users
- locale – Get locale specific information
- localedef – Compile locale definitions
- logname – Print user's login name
- man – Manual browser
- mesg – Control write access to your terminal
- passwd – Change user password

User management and support

- su – Start a new process (defaults to shell) as a different user (defaults to root)
- sudo – execute a command as a different user.
- users – Show who is logged on (only users names)
- w – Show logged-in users and their current tasks
- whatis – command description from *whatis* database
- whereis – locates the command's binary and manual pages associated with it
- which (Unix) – locates where a command is executed from
- who – Show who is logged on (with some details)
- write – Send a message to another user