UNIX®

A Standard of The Open Group®

Computer Industry Evolution: From Turing to Unix

Dr.B.W.Stuck

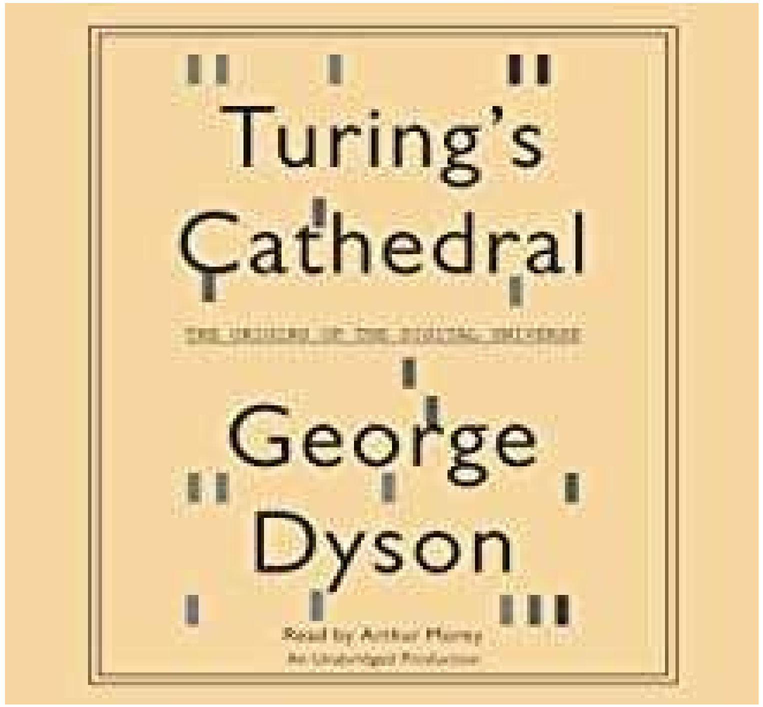
MIT: SBEE 1968, SMEE 1969, EE 1970, PhD 1972

Google Scholar profile https://scholar.google.com/citations?user=ykewuCAAAAAJ&hl=en

Outline

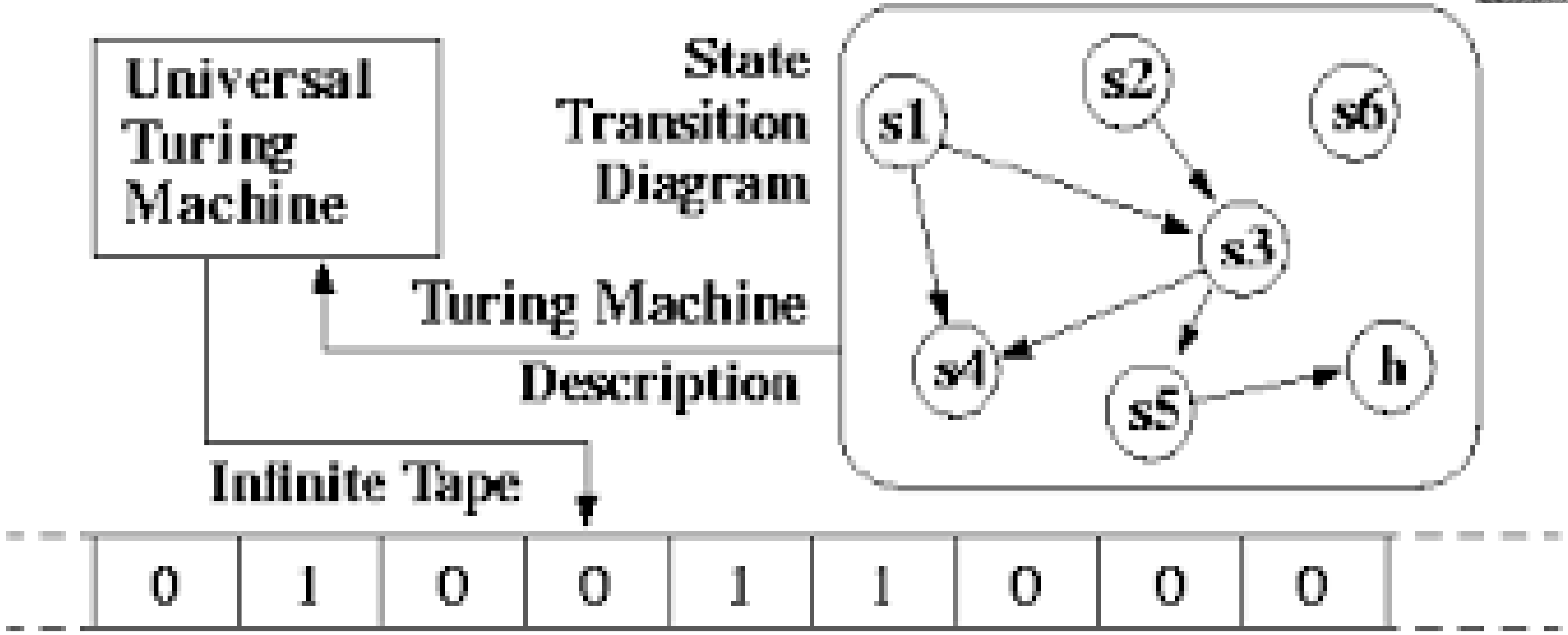
- The 1940s: setting the foundations
- 1950s: the first commercial electronic computers
- 1960s: IBM launches S/360 mainframes
- 1970s: DEC, HP, Wang, Fujitsu, Prime, NEC launch minicomputers
- 1980s: Apple, IBM, Commodore, Tandy personal computers
- What About Software?
- Development of Unix
- Summary

1940s: Setting the Foundations of Digital Computing

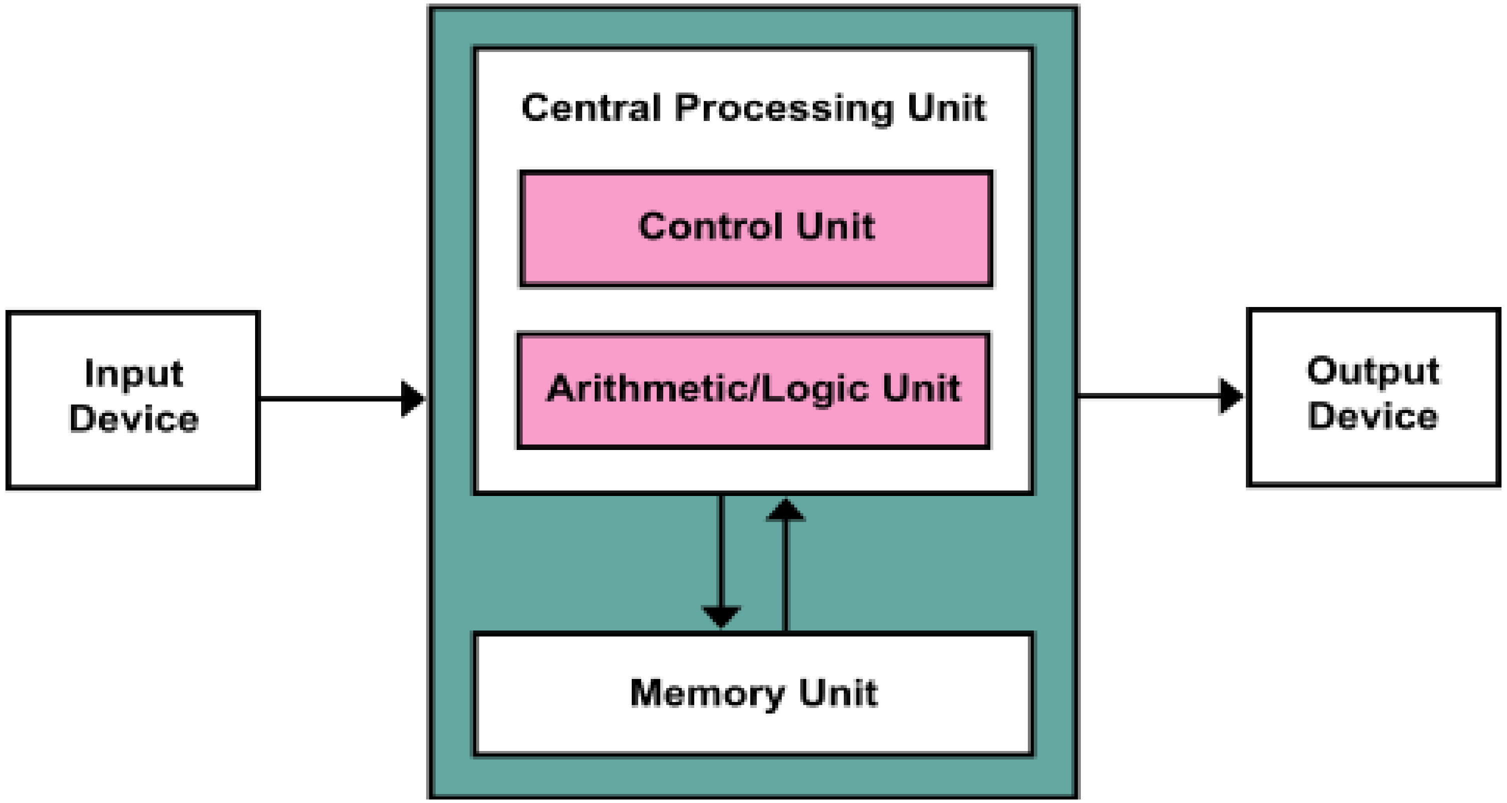


Note: A Turing machine is a mathematical model of computation describing an abstract machine that manipulates symbols on a strip of tape according to a table of rules. Despite the model's simplicity, it can implement any computer algorithm.

What Makes Up a Computer: Turing Model



What Makes Up a Computer: von Neumann Model



1940s: Setting the Foundations

- Impetus from military: artillery, radar, nuclear bombs
- Different groups in USA and Western Europe: von Neumann at Institute for Advanced Studies and the IBM 650
- Basic hardware modules
  - Processor
  - Memory (hierarchy based on economics)
  - Various input/output devices: printer, modem, terminal
  - Switch fabric/bus interconnection
- Stored programs written in machine language/assembly language

1950s: Commercialization of Digital Computers

- Basic hardware technology
  - vacuum tubes for flip flops and gates in processor
  - Magnetic core storage for processor
  - Magnetic drum/disk drive and tape and punched hole paper for secondary storage
- Software: FORTRAN, BASIC, LISP, early operating systems
- Commercial vendors
  - IBM 650,704, 709, 7094, 7094 Stretch
  - Burroughs: B5500
  - Univac
  - Control Data CDC6600
  - General Electric GE645
  - Bull
  - ICL
  - Fujitsu

1960s: Commercialization of Digital Computers

- Basic hardware technology
  - transistors for flip flops and gates in processor
  - Magnetic core storage for processor-> static and dynamic random access memory
  - Magnetic disk driv/tape for secondary storage
  - Terminals and front end processors to handle communications input/output
- Software: PL/1, Fortran, Cobol, Information Management System (IMS), CICS, OS 360, MULTICS
- Commercial vendors
  - IBM S/360
  - Burroughs: B6500
  - Univac
  - Control Data
  - General Electric
  - Bull
  - ICL
  - Fujitsu

Interfaces and Architecture

- Digital systems showed the importance of a well defined interface between different modules
  - Provided necessary modularity
  - Allowed unit testing before system integration testing
  - Speeded up fault detection and isolation and recovery
  - Provided for incredible complexity
- Architecture is the specification of all interfaces between all subsystems
  - Ideally each subsystem is of equivalent comparable complexity
  - Allows for better estimates of staffing, resources and time to develop each subsystem

1970s: Minicomputers

- Basic hardware technology
  - transistors for flip flops and gates in processor
  - Transistor based storage for processor
  - Magnetic drum/disk drive for secondary storage
  - Terminals and graphics support
- Software: C, UNIX, vendor proprietary operating systems
- Commercial vendors
  - IBM Series 1, System 34/38
  - Digital Equipment Corporation
  - Hewlett Packard
  - Wang Laboratories
  - Prime
  - Fujitsu
  - NEC

1980s: Personal Computers

- Basic hardware technology
  - Intel 8086, Motorola 68000, Zilog Z8000 processors
  - Transistor based static and dynamic random access memory
  - Magnetic disk drive/tape for secondary storage
  - Terminals and graphics support, mouse, keyboard
  - Client/server: servers in cloud, servers/clients connected to network
- Software: CPM, DOS, UNIX, Visicalc, Wordstar, Ashton Tate, Oracle
- Commercial vendors
  - Apple I, Apple II, Macintosh
  - Comordore
  - Tandy
  - IBM Personal Computer
  - Dell, CompuAdd, Lenovo

Personal Computer: New Business Model

- Open standards led to multiple parties introducing personal computer hardware and software
- Everyone was using the same PC bus
- Multiple companies introduced add-on boards and software
- Everyone was using the same operating system (PC-DOS/MS-DOS)
- Everyone was using the same processor chip set (mostly Intel)
- Intel and Microsoft combined provided the real value in the computer industry

Waves of Product Developments

Category	Peak Growth Rate Year in Shipments
Mainframe	1965
Minicomputer	1972
Personal Computer	1979
Workstation	1986
Mobile Phone	1993

Innovations That Changed Computer Technology

- Microprocessors
  - Notable turning point in Computer History allowing tens to hundreds of billions of transistors in a small chip.
- General-purpose GPU (GPGPU)
  - GPU Allows very time consuming algorithms to run fast.
- Fortran
  - Fortran was the first widely-used high-level general-purpose programming language.
- Unix Operating System
  - All modern operating systems are in a way derivative of Unix.
- Nexus
  - Tim Berners-Lee of CERN implemented the WorldWideWeb web browser (Nexus) in 1990.

https://medium.com/swlh/top-5-amazing-innovations-that-changed-computer-technology-5fd49475e44b

What About Software?

- First operating system developed at Bell Labs to improve application development productivity at the expense of using processor cycles and memory (V Vyssotsky and R Hamming for IBM 650)
- DARPA funded MULTICS in 1964 (joint effort with MIT, Bell Labs, and General Electric for GE-645)
- Ken Thompson develops UNIX, stripped version of MULTICS, in 1969-1970 on PDP 11/45, all programmed in assembler
- Dennis Ritchie develops C programming language, UNIX reimplemented in C in 1973 and released
- Ken Thompson takes a take with a copy of UNIX to University of California/Berkeley in 1975 and Bill Joy introduces Berkeley version of UNIX
- Procedural programming -> object oriented programming

UNIX Evolution

UNIX History

UNIX®

A Standard of The Open Group®

- 1964-1969: Massachusetts Institute of Technology,Bell Labs, and General Electric were developing Multics, a time-sharing operating system for the GE-645 mainframe computer. Multics featured several innovations, but also presented severe problems with memory .Bell Labs started withdrawing from the project.
- Bell Labs pulled out 1969: Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna, who decided to reimplement their experiences anew on smaller scale.
- The new operating system was a single-tasking system. In 1970, the group coined the name Unics for **Uniplexed Information and Computing Service** as a pun on Multics, which stood for **Multiplexed Information and Computer Services**. Brian Kernighan takes credit for the idea.
- UNIX originally written in assembly language, but in 1973, Version 4 Unix was rewritten in C. Version 4 Unix, however, still had many PDP-11 dependent codes, and was not suitable for porting. The first port to another platform was made five years later (1978) for the Interdata 8/32.

UNIX History

UNIX®

A Standard of The Open Group®

- Bell Labs produced several versions of Unix that are collectively referred to as Research Unix. In 1975, the first source license for UNIX was sold to Donald B. Gillies at the University of Illinois Urbana–Champaign Department of Computer Science (UIUC). UIUC graduate student Greg Chesson, who had worked on the Unix kernel at Bell Labs, was instrumental in negotiating the terms of the license.
- 1975-1985: large-scale adoption of Unix (BSD and System V) by commercial startups, which led to Unix fragmenting into multiple, similar but often slightly mutually-incompatible systems including DYNIX, HP-UX, SunOS/Solaris, AIX, and Xenix. In the late 1980s, AT&T Unix System Laboratories and Sun Microsystems developed System V Release 4 (SVR4), which was subsequently adopted by many commercial Unix vendors.
- In the 1990s, Unix and Unix-like systems grew in popularity and became the operating system of choice for over 90% of the world's top 500 fastest supercomputers.
- In 2000, Apple released Darwin, also a Unix system, which became the core of the Mac OS X operating system, later renamed macOS. iOS is a derivative of UNIX, as is Android.
- Unix-like operating systems are S used in modern servers, workstations, and mobile devices.

UNIX Software Overview

UNIX®

A Standard of The Open Group®

- Unix systems: modular design is "Unix philosophy".
- **The operating system should provide a set of simple tools, each of which performs a limited, well-defined function.**
- A unified and inode-based filesystem (the Unix filesystem)
- An inter-process communication mechanism known as "**pipes**" serve as the main means of communication
- shell scripting and command language (the Unix shell) is used to combine the tools to perform complex workflows.
- Unix distinguishes itself from its predecessors as the first portable operating system: almost the entire operating system is written in the C programming language, which allows Unix to operate on numerous platforms.

UNIX Software Overview

A Standard of The Open Group®

- Unix was originally designed by software experts to provide a world class software development environment
- At first, Unix was not designed to be portable or for multi-tasking.
- Unix gradually gained portability, multi-tasking and multi-user capabilities in a time- sharing configuration.
- The UNIX philosophy: the use of plain text for storing data; a hierarchical file system; treating devices and certain types of inter-process communication (IPC) as files; and the use of a large number of software tools, small programs that can be strung together through a command-line interpreter using pipes, as opposed to using a single monolithic program that includes all of the same functionality.

UNIX Standards

UNIX®

A Standard of The Open Group®

- In the late 1980s, an open operating system POSIX provided a common baseline for all operating systems; IEEE based POSIX around the common structure open systems
- In the early 1990s, a separate effort, the Common Open Software Environment (COSE) initiative, which eventually became the Single UNIX Specification (SUS) administered by The Open Group.
- In 1999, in an effort towards compatibility, several Unix system vendors agreed on SVR4's Executable and Linkable Format (ELF) as the standard for binary and object code files. The common format allows substantial binary compatibility among different Unix systems operating on the same CPU architecture.
- The Filesystem Hierarchy Standard was created to provide a reference directory layout for Unix-like operating systems; it has mainly been used in Linux.

UNIX Structure

UNIX®

A Standard of The Open Group®

- Kernel
- Development environment
- Commands
- Documentation
- Tools

UNIX Kernel Overview

The kernel provides

- services to start and stop programs,
- handles the file system and other common "low-level" tasks that most programs share, and
- schedules access to avoid conflicts when programs try to access the same resource or device simultaneously.
- To mediate such access, the kernel has special rights, reflected in the distinction of kernel space from user space, the latter being a priority realm where most application programs operate.

UNIX Kernel Overview

UNIX®

A Standard of The Open Group®

Kernel – source code in /usr/sys, composed of several sub-components:

- conf – configuration and machine-dependent parts, including boot code
- dev – device drivers for control of hardware (and some pseudo- hardware)
- sys – operating system "kernel", handling memory management, process scheduling, system calls, etc.
- h – header files, defining key structures within the system and important system-specific invariables

UNIX Development Environment

UNIX

A Standard of The Open Group®

Development environment – early versions of Unix contained a development environment sufficient to recreate the entire system from source code:

- ed – text editor, for creating source code files
- cc – C language compiler (first appeared in V3 Unix)
- as – machine-language assembler for the machine
- ld – linker, for combining object files
- lib – object-code libraries (installed in /lib or /usr/lib). libc, the system library with C run-time support, but always additional libraries for things such as mathematical functions (libm) or database access. V7 Unix introduced the first version of the modern "Standard I/O" library stdio

UNIX Development Environment

UNIX®

A Standard of The Open Group®

- I/O library stdio as part of the system library. Later implementations increased the number of libraries significantly.
- make – build manager (introduced in PWB/UNIX), for effectively automating the build process
- include – header files for software development, defining standard interfaces and system invariants
- Other languages – V7 Unix contained a Fortran-77 compiler, a programmable arbitrary-precision calculator (bc, dc), and the awk scripting language; later versions and implementations contain many other language compilers and toolsets. Early BSD releases included Pascal tools, and many modern Unix systems also include the GNU Compiler Collection as well as or instead of a proprietary compiler system.
- Other tools – including an object-code archive manager (ar), symbol-table lister (nm), compiler-development tools (e.g. lex & yacc), and debugging tools.

UNIX Commands

A Standard of The Open Group®

Commands – Unix makes little distinction between commands (user-level programs) for system operation and maintenance (e.g. cron), commands of general utility (e.g. grep), and more general-purpose applications such as the text formatting and typesetting package. Nonetheless, some major categories are:

- sh – the "shell" programmable command-line interpreter, the primary user interface on Unix before window systems appeared, and even afterward (within a "command window").
- Utilities – the core toolkit of the Unix command set, including cp, ls, grep, find and many others. Subcategories include:
- System utilities – administrative tools such as mkfs, fsck, and many others.
- User utilities – environment management tools such as passwd, kill, and others.

UNIX Commands

UNIX®

A Standard of The Open Group®

- Document formatting – Unix systems were used from the outset for document preparation and typesetting systems, and included many related programs such as nroff, troff, tbl, eqn, refer, and pic. Some modern Unix systems also include packages such as TeX and Ghostscript.
- Graphics – the plot subsystem provided facilities for producing simple vector plots in a device-independent format, with device-specific interpreters to display such files. Modern Unix systems also generally include X11 as a standard windowing system and GUI, and many support OpenGL.
- Communications – early Unix systems contained no inter-system communication, but did include the inter-user communication programs mail and write. V7 introduced the early inter-system communication system UUCP, and systems beginning with BSD release 4.1c included TCP/IP utilities.

UNIX Documentation

A Standard of The Open Group®

Documentation – Unix was one of the first operating systems to include all of its documentation online in machine-readable form. The documentation included:

- man – manual pages for each command, library component, system call, header file, etc.
- doc – longer documents detailing major subsystems, such as the C language and troff

UNIX Impact

UNIX®

A Standard of The Open Group®

- The Unix system had a significant impact on other operating systems by its interactivity, by providing the software at a nominal fee for educational use, by running on inexpensive hardware, and by being easy to adapt and move to different machines. Unix was originally written in assembly language, but was soon rewritten in C, a high-level programming language.
- Unix had a drastically simplified file model compared to many contemporary operating systems: treating all kinds of files as simple byte arrays. The file system hierarchy contained machine services and devices (such as printers, terminals, or disk drives), providing a uniform interface, but at the expense of occasionally requiring additional mechanisms such as ioctl and mode flags to access features of the hardware that did not fit the simple "stream of bytes" model.

UNIX Impact

UNIX®

A Standard of The Open Group®

- Unix popularized the hierarchical file system with arbitrarily nested subdirectories, originally introduced by Multics. Other common operating systems had ways to divide a storage device into multiple directories or sections, but with a fixed number of levels, often only one. Several major proprietary operating systems eventually added recursive subdirectory capabilities also patterned after Multics. DEC's RSX-11M's "group, user" hierarchy evolved into OpenVMS directories, CP/M's volumes evolved into MS-DOS 2.0+ subdirectories, and HP's MPE group.account hierarchy and IBM's SSP and OS/400 library systems were folded into broader POSIX file systems.
- Making the command interpreter an ordinary user-level program, with additional commands provided as separate programs, was another Multics innovation popularized by Unix. The Unix shell used the same language for interactive commands as for scripting (shell scripts – there was no separate job control language like IBM's JCL). Since the shell and OS commands were "just another program", the user could choose (or even write) their own shell. New commands could be added without changing the shell itself. Unix's innovative command-line syntax for creating modular chains of producer-consumer processes (pipelines) made a powerful programming paradigm (coroutines) widely available.

UNIX Impact

UNIX®

A Standard of The Open Group®

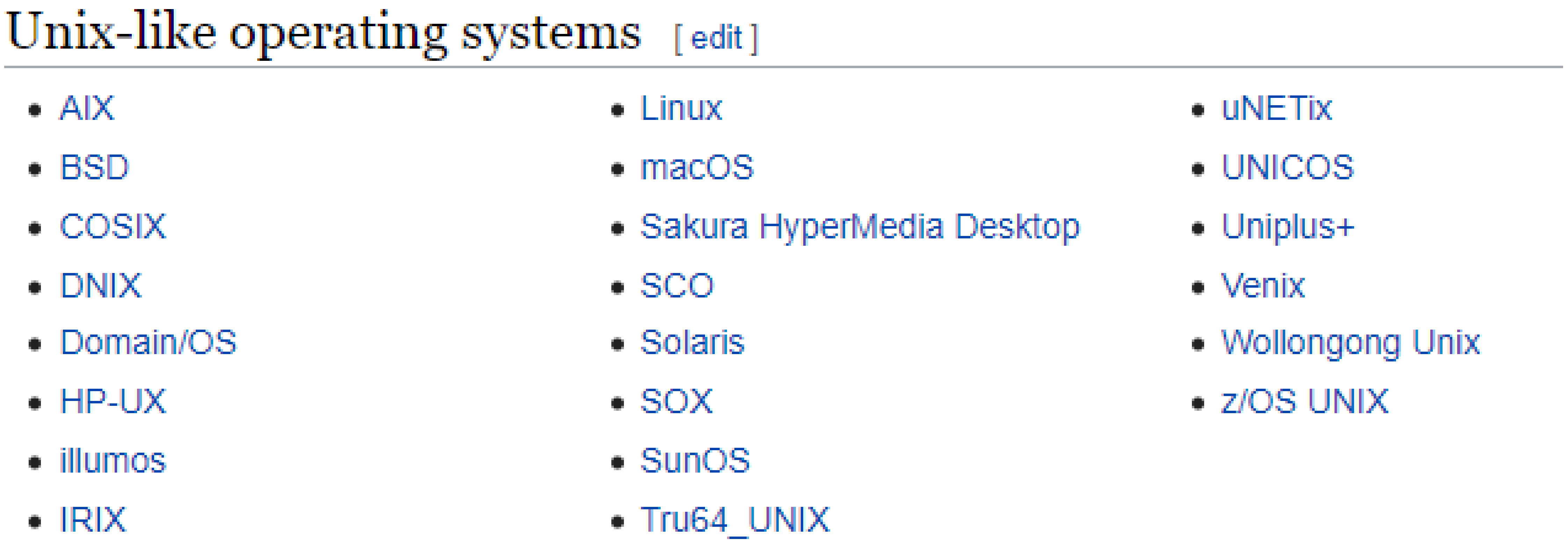
- A fundamental simplifying assumption of Unix was its focus on newline-delimited text for nearly all file formats. There were no "binary" editors in the original version of Unix – the entire system was configured using textual shell command scripts. The common denominator in the I/O system was the byte – unlike "record-based" file systems. The focus on text for representing nearly everything made Unix pipes especially useful and encouraged the development of simple, general tools that could be easily combined to perform more complicated ad hoc tasks. The focus on text and bytes made the system far more scalable and portable than other systems. Over time, text-based applications have also proven popular in application areas, such as printing languages (PostScript, ODF), and at the application layer of the Internet protocols, e.g., FTP, SMTP, HTTP, SOAP, and SIP.
- Unix popularized a syntax for regular expressions that found widespread use. The Unix programming interface became the basis for a widely implemented operating system interface standard (POSIX). The C programming language soon spread beyond Unix, and is now ubiquitous in systems and applications programming.

UNIX Impact

UNIX®
A Standard of The Open Group®

- Early Unix developers were important in bringing the concepts of modularity and reusability into software engineering practice, spawning a "software tools" movement. Over time, the leading developers of Unix (and programs that ran on it) established a set of cultural norms for developing software
- The TCP/IP networking protocols were quickly implemented on the Unix versions widely used on relatively inexpensive computers, which contributed to the Internet explosion of worldwide real-time connectivity, and which formed the basis for implementations on many other platforms.
- The Unix policy of extensive on-line documentation and (for many years) ready access to all system source code raised programmer expectations, and contributed to the launch of the free software movement in 1983.

Unix Like Operating Systems



Unix-like operating systems [edit]

- AIX
- BSD
- COSIX
- DNIX
- Domain/OS
- HP-UX
- illumos
- IRIX
- Linux
- macOS
- Sakura HyperMedia Desktop
- SCO
- Solaris
- SOX
- SunOS
- Tru64_UNIX
- uNETix
- UNICOS
- Uniplus+
- Venix
- Wollongong Unix
- z/OS UNIX

C Programming Language

- C is an imperative procedural language supporting structured programming, lexical variable scope, and recursion, with a static type system.
- It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming.
- A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code.

Design Point Concepts

- There are two fundamental questions in computing
  - Does the data get sent to the program for processing?
  - Does the program go to the data for processing?
- A fundamental issue in computers is the address space, how many bits are available to address items? Initially memory was very expensive, and address space was eight bits, then twelve bits, then sixteen bits, then thirty two bits, and on and on
  - What drives the increase in address space is a statistical observation: based on analyzing thousands of application programs, the address space increases by 1.5 bits per year, since the start of commercial computing

Summary

- The computer industry has rapidly evolved over the past eight decades to impact virtually every aspect of daily life, building on wave after wave of technology advances that increased performance for every decreasing price
- The underlying hardware technology based on solid state electronics saw the evolution from one transistor on a chip to billions
- The underlying software technology has transitioned from procedural to object oriented programming, with initial programs consisting of a thousand lines of code evolving to systems of tens of millions of lines of code
- The Internet started from two computers in 1969 to over 8 billion addressable nodes today

B.W.Stuck Biography

- MIT, 1964-1972: SBEE 1968, SMEE 1969, EE 1970, PhD 1972
- Bell Laboratories 1972-1984:
  - Lectured at over 50 universities and research institutes in North America, Western Europe, Soviet Union, Japan.
  - Worked on 20 UNIX application systems, consulted on another 80, created a class for a masters level computer science program, which was taught at Columbia University three times, leading to publishing a book, A Computer and Communication Network Performance Analysis Primer, Prentice Hall, 1985
- Independent consulting, 1984-1998
- Venture capital, 1998-today