INTRO TO OPERATING SYSTEMS

OS PAST/PRESENT/THEMES/ERAS & FUTURE/RESEARCH

Evolution of Operating Systems

- The evolution of operating systems is directly dependent to the development of computer systems and how users use them. Here is a quick tour of computing systems through the past fifty years in the timeline.

https://ppt-online.org/441021

THIS WEEK'S 2x75min 4p-5:15p

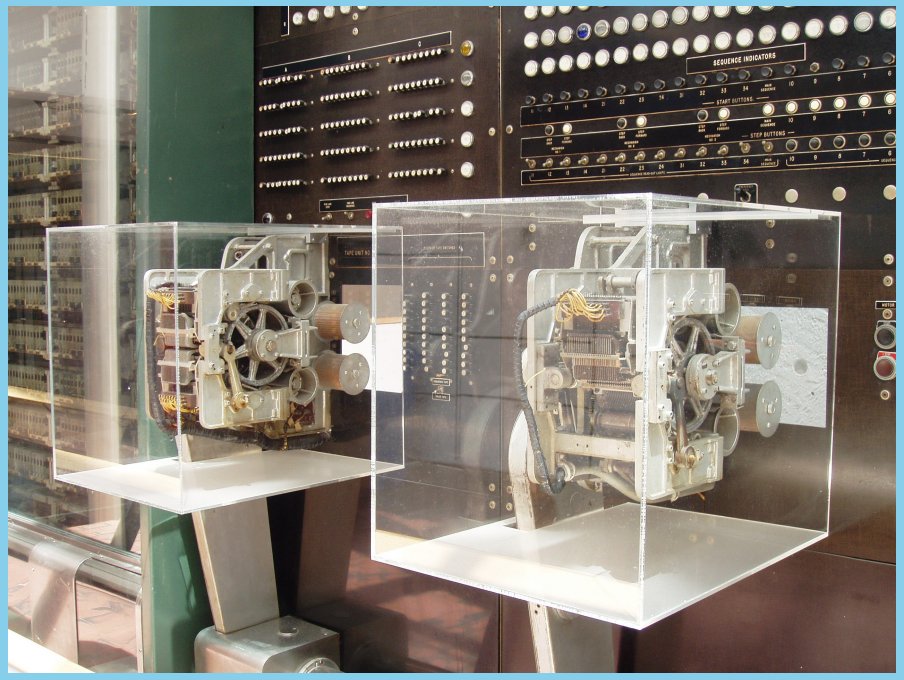
♦ (Carryover) Whys and What Ifs (15m)

♦ (Carryover) Some Architectures Needing an OS (15m)

- Issues as They Arose (30m)
- Survey of the Present (30m)
- Recent Research Titles (30m)
- Discussion of Lab I (30m)

♦ Issues as They Arose (20m)

♣ Program Loading: Mark I



http://davidad.github.io/blog/2014/03/12/the-operating-system-is-out-of-date/

♣ Hardware Abstraction: JCL

For example, to copy a file on Unix operating system, the user would enter a command like:

```
cp oldFile newFile
```

The following example, using JCL, might be used to copy a file on OS/360:

```
//IS198CPY JOB (IS198T30500),'COPY JOB',CLASS=L,MSGCLASS=X
//COPY01   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=OLDFILE,DISP=SHR
//SYSUT2   DD DSN=NEWFILE,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(40,5),RLSE),
//            DCB=(LRECL=115,BLKSIZE=1150)
//SYSIN  DD DUMMY
```

https://en.wikipedia.org/wiki/Job_Control_Language

♣ Command Set for Files, Programs, Status: CPM/DOS

CP/M PLUS COMMAND SUMMARY .....

DOS commands

http://www.cpm.z80.de/manuals/cpm3-cmd.pdf https://sites.google.com/site/pcdosretro/commands

---

♣ Command Set for Files, Programs, Status: TRS80

TRS-80.org

Search

Computers Arcade Games Interviews

Hardware Books Software Magazines

Model I TRSDOS Commands

written by Matthew Reed

The Model I TRSDOS command shell supported a number of internal commands, as documented by the TRSDOS 2.3 Reference Manual. Most of the commands were also supported

http://www.trs-80.org/trsdos-model1-commands/

♣ Command Set for Files, Programs, Status: Unix v7

♣ Command Set for Files, Programs, Status: CTSS

>

Search The UNIX and Linux Forums (Quick Search)

The UNIX and Linux Forums

Forums Man Search Today's Posts Quick Links Register

Repository Loaded

Unix Version 7 Commands - 310 Total Man Pages in Repository

https://www.unix.com/man-page-v7-repository.php

♣ Command Set for Files, Programs, Status: Embedded OS

Kernel [edit]

CTSS used a modified IBM 7090 mainframe computer that had two 32,768 (32K) 36-bit-word banks of core memory instead of the normal one.[11] One bank was reserved for the time-sharing supervisory program, the other for user programs. CTSS had a protected-mode kernel, the supervisor's functions in the A-core (memory bank A) could be called only by software interrupts, as in the modern operating systems. Causing memory protection interrupts were used for software interrupts.[4] Processor allocation scheduling with equal time-slit 100 ms was controlled by a multilevel feedback queue. CTSS also had some special hardware, a clock interrupt, the "pulse" to up certain housekeeping

Supervisor subroutines [edit]

- RDFLXA – Read an input line from console
- WRFLX – Write an output line to console
- DEAD – Put the user into dead status, with no program in memory
- DORMNT – Put the user into dormant status, with program in memory
- GETMEM – Get the size of the memory allocation
- SETMEM – Set the size of the memory allocation
- TSSFIL – Get access to the CTSS system files on the disk
- USRFIL – Change back to user's own directory
- GETBRK – Get the instruction location counter at quit

Disk-control subroutines [edit]

- .DUMP – Dump a continuous block onto file
- .LOAD – Load a continuous block from file
- .ASIGN – Prepares file for writing
- .APEND – Prepares file for appending
- .SEEK – Prepares file for reading
- .RELRW – Prepares file for reading and writing
- .WRITE – Write data to a relative location in file
- .READK – Read data from a relative location in file
- .FILC – Terminate writing of file
- .ENDRD – Terminate reading of file
- .DLETE – Delete a file
- .RENAM – Rename a file and change its mode
- .FILDR – Obtain a copy of the user file directory
- .FSTAT – Get information about a file

Console commands [edit]

- login – Log into system
- logout – Log out of system
- listf – List files in the directory
- input – Input source code, fixed size lines
- edit – Edit source code in a BASIC style with line numbers
- printf – Print file starting from a line number
- fap – FAP assembler
- mad – MAD compiler
- madtrn – Fortran II to MAD translator
- load – Load binaries (linking in memory)
- use – Load missing binaries
- start – Run program loaded into memory
- save – Save program in the memory to file
- resume – Load saved program and resume running it
- pm – Get post-mortem information of the program in memory
- patch – Edit memory
- tra – Create transfer to a relative location in a program
- stopat – Create transfer to stop the program at a location
- rename – Rename file
- chmode – Change the mode of the file
- delete – Delete file, has wildcards
- split – Split file
- combin – Join files, also binary files, making libraries
- cpu – Get the current machine conditions
- octlk – Print memory
- memo – Input text files, variable size lines
- modify – Edit text files, similar to edit
- ditto – Print text files with formatting (footnotes, pages)

https://e

The true descendant of the early operating systems is what is now called the "kernel". In technical and development circles the old restricted sense of an OS persists because of the continued active development of embedded operating systems for all kinds of devices with a data-processing component, from hand-held gadgets up to industrial robots and real-time control-systems, which do not run user applications at the front-end. An embedded OS in a device today is not so far removed as one might think from its ancestor of the 1950s.

https://en.wikipedia.org/wiki/History_of_operating_systems

♣ Multitasking/Time-Sharing: Processes, Interrupts, Scheduling

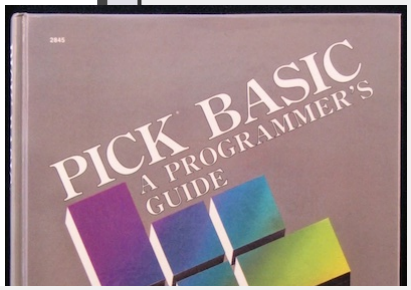
♣ Memory Hierarchy: Swapping, File Systems: VMS/Pick

https://en.wikipedia.org/wiki/Computer_multitasking https://en.wikipedia.org/wiki/Time-sharing

Search

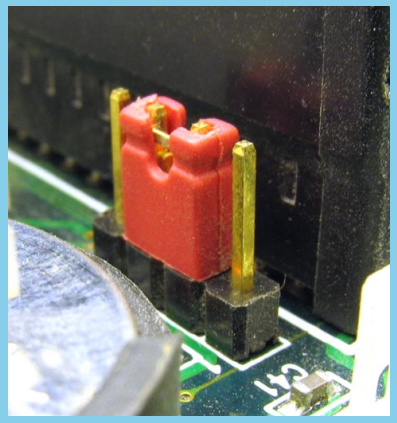
Jonathan E. Sisk's Pick/BASIC: A Programmer's Guide

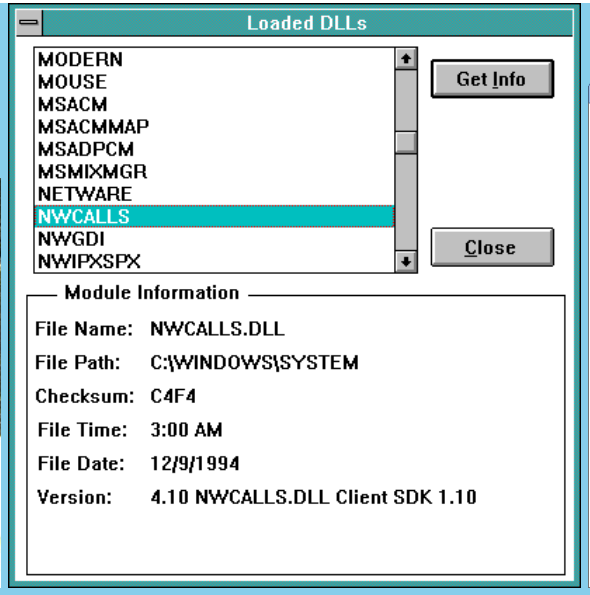
Chapter 1 - Pic

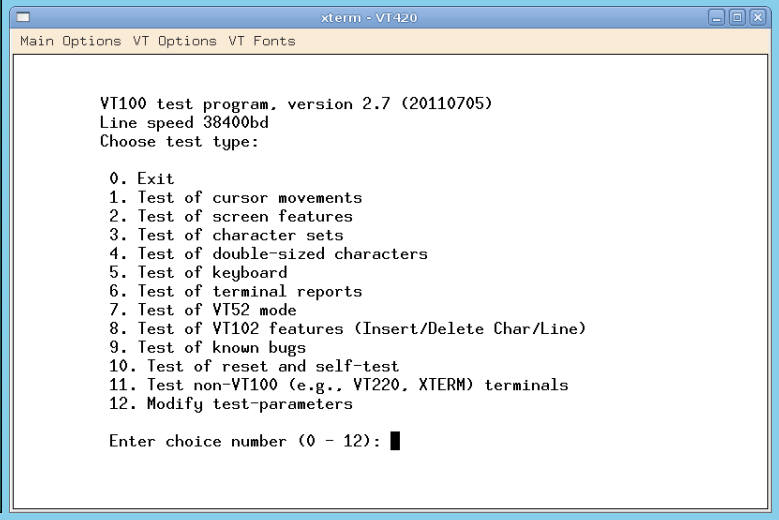


♣ Peripherals, GUI: Configuration, Emulation

https://users.soe.ucsc.edu/~sbrandt/221/Papers/Memory/levy-computer82.pdf https://secure28.securewebsession.com/jes.com/pb/pb_wp1.html







http://teraknorblogs.blogspot.com/2015/12/setting-them-irqs-and-other-long-lost.html
https://support.novell.com/techcenter/articles/ana19950502.html https://invisible-island.net/vttest/

♣ COTS Networking, Audio, Video: I/O Drivers

https://msfn.org/board/topic/107001-compatible-hardware-with-windows-9x/

♣ Users

Contention, Switching, Usability, Responsiveness, Protection, Privileges, Monitoring

♣ Devices

I/O Efficiency, Interfaces, Interrupts, Queues, Block Transfer / DMA

♣ Parallel and Distributed Computing

Remote Procedure Calls, Synchronization, Consistency, Speedup, Shared Memory

---

♣ Networking, Internet

Servers / Clients, Logging, Rich Media / Streams* / Data Warehousing, Security / Hacking

♣ Mobility

Keyboard?, CPU Stepping / Battery, Connectivity, Authentication

♣ Clouds

Virtual Machines, Scalability, Orchestration, Heat* / Space*

---

* Original Concerns

♦ Survey of the Past (15m)

♣ 1950s: ERA OF HOPE

Not many programs, programmers, devices, Not much data

Lots of vision, patience

♣ 1960s: ERA OF IBM BUSINESS AND LEAD UNIVS

Not much memory, slow I/O

Lots of conceptual invention, esp. programming languages, discipline

---

♣ 1970s: ERA OF SCIENTISTS AND HOBBYISTS

Not much memory, slow cpus, not many commands, inconvenient secondary storage

Lots of great future programmers on hobby machines, cleverness

Xerox Alto (1973): 5.8MHz, 0 cache, 128K RAM, 1.5M/s

Apple II MOS 6502 (1977): 1MHz, 0 cache, 4K-48K RAM, EPP 2M/s

---

♣ 1980s: ERA OF OWNERSHIP

Not much memory, slow cpus, not much disk, networking awful

Lots of personal computers, micro computers, gui cpu-intensive

IBM PC XT I 8088 (1983): 5MHz, 4B prefetch q, 640K RAM, ISA 8M/s

---

♣ 1990s: ERA OF GOLD

Not much memory, bandwidth

Lots of nodes, data, people

DEC Alpha 21164PC (1995): 533MHz, 8K/96K/2M L0/L1/L2, 1G RAM, PCI 264M/s

---

♣ 2000s: ERA OF PLENTY

Vision for use, not device

Lots of cpus, lots of memory, lots of storage, lots of consolidation

HP Elitebook 6930p I T9600 (2008): 2x2.8GHz, ?/4x32K/6M L0/L1/L2, 8G RAM, SATA II 375M/s

Samsung S3 C A9 (2012): 4x1.4GHz, ?/2x34K/8M L0/L1/L2, 2G RAM, UHS II 156-312M/s

Units are in 8-bit Bytes, eg, GB, MB, KB

♦ Survey of the Present (15m)

♣ 2010s: ERA OF WASTE

Not many hardware issues, too much data, people problems

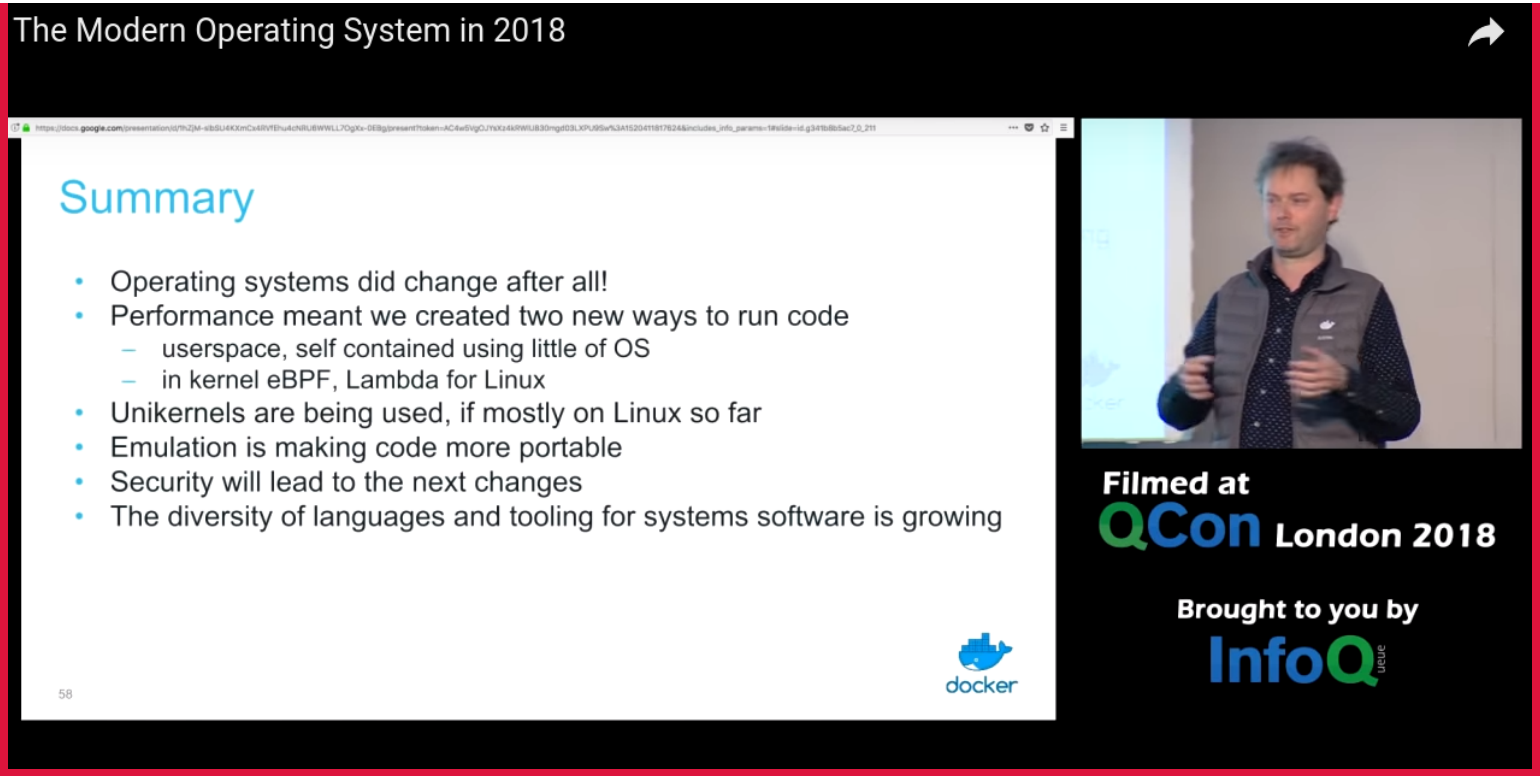
Lots of different devices for segmented markets, virtual machines

IBM BM+ I X6248 (2019): 40x2.5GHz, 0K/2x32K/1M L0/L1/L2, 32G RAM, SATA III 600M/s

♣ Issues

Scalability, Flexibility, Deployment, Library Minimization, Real-Time, IOT Embedded

---



https://www.youtube.com/watch?v=dR2FH8z7L04

♣ 2020s: ERA OF DEPENDENCE

Sensors, social regulation, privacy and analytics, epistemics and cyber, horses/cars/devices

♣ Issues

Longevity, Archiving and Control, Data Integrity, Infrastructure, Robustness, Migration, Pricing

♣ Predictions

♥ OS will manage pseudonymization, noise injection and randomization, sandboxing, containerization

♥ OS will manage robust consent and disconnection, erase-and-forget-me-apis

♥ OS will manage backup, inter-device data transfer, availability

Recent Research Titles (30m)

♣ 2015

- *Network operating system profiling and monitoring using network data integrity* KE Amidon, PJ Balland III, N Gude… - **US Patent** …, 2015
- *System and method for providing a secured operating system execution environment* AS Saltan - **US Patent** 9,087,199, 2015
- *Operating system supporting cost aware applications* P Menezes, M Piumatti, UW Parks, R Rao - **US Patent** 8,971,841, 2015
- *Determining compatibility of an application with different versions of an operating system* V Bhat - **US Patent** 9,015,702, 2015

♣ 2016

- *Arrakis: The operating system is the control plane* S Peter, J Li, I Zhang, DRK Ports, D Woos… - ACM Transactions on …, 2016
- *Vehicle comprising multi-operating system* CP Ricci, B Reeves, PE Reeves, R Teltz… - **US Patent** …, 2016
- *Approaches for protecting sensitive data within a guest operating system* G Tedesco, A Pole, A Southgate, I Pratt… - **US Patent** …, 2016
- *Data mover permitting data transfer without transferring data between application and operating system* SB Vaghani, M Rawat, RAI Abhishek -

US Patent 9,454,368, 2016

♣ 2017

- *Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system* M Kamel, T Stastny, K Alexis, R Siegwart - Robot operating system (ROS), 2017
- *Cloud service enabled to handle a set of files depicted to a user as a single file in a native operating system* G Dorman, S Asok, M Self - **US Patent** 9,575,981, 2017
- *Operating system patching and software update reconciliation* AC Steigleder - **US Patent** 9,766,873, 2017
- *Security for the robot operating system* B Dieber, B Breiling, S Taurer, S Kacianka… - Robotics and …, 2017

♣ 2018

- *Hyperledger fabric: a distributed operating system for permissioned blockchains* E Androulaki, A Barger, V Bortnikov, C Cachin… - Proceedings of the …, 2018
- *Vehicle operating system using motion capture* JC Hsiao, YS Chen - **US Patent** App. 10/124,648, 2018
- *Training an at least partial voice command system* DW Pitschel, AJ Cheyer, CD Brigham… - **US Patent** …, 2018
- *RIOT: An open source operating system for low-end embedded devices in the IoT* E Baccelli, C Gündoğan, O Hahm… - IEEE Internet of …, 2018

♣ 2019

- *Personal health operating system* P Soon-Shiong, V Rangadass… - **US Patent** App. 10 …, 2019
- *Operating system support for game mode* GJ Colombo, L Seetharaman, G Wong… - **US Patent** App. 10 …, 2019
- *SPIN: Seamless operating system integration of peer-to-peer DMA between SSDs and GPUs* S Bergman, T Brokhman, T Cohen… - ACM

Transactions on …, 2019

- *Thunderclap: Exploring Vulnerabilities in Operating System IOMMU Protection via DMA from Untrustworthy Peripherals.* AT Markettos, C Rothwell, BF Gutstein, A Pearce… - NDSS, 2019

♦ "Operating Systems" at scholar.google Since 2015?

♦ Can a Browser be an Essential Part of an OS?

♣ Antitrust

♣ deletion of any file containing browsing-specific routines would also delete vital operating system routines

♣ Testimony

♣ Rant

♣ Irony

♦ Can changing one constant be a major update to an OS?

♣ Investigation

- ♣ Likes
- ♣ Dislikes
- ♣ Statement
- ♣ Analysis
- ♣ between two and four weeks work for six to ten engineers, assuming a consistent and motivated workforce

♦ Discussion of Lab 1 (10m)

- ♣ Timeslicing writers?
- ♣ Queue priority fairness / Cheating the rules?
- ♣ Requesting memory
- ♣ Seeing queue sizes / What is the value of information?
- ♣ Implementing priority

♦ Discussion of Lab 2 (10m)

- ♣ What if there were 8 instead of 4 processes?
- ♣ What if processes were truly blind?
- ♣ What if processes were random-window ready to acquire?
- ♣ What if each process had a deadline?
- ♣ What if two teams were seated instead of one?

♦ Discussion of HW 1 (10m)

- ♣ Queueing vs Queuing?
- ♣ Kleinrock and CWRU? VII vs ed2
- ♣ Something about Purdue
- ♣ Ozsoyoglus
- ♣ Attention to detail, Attempting all