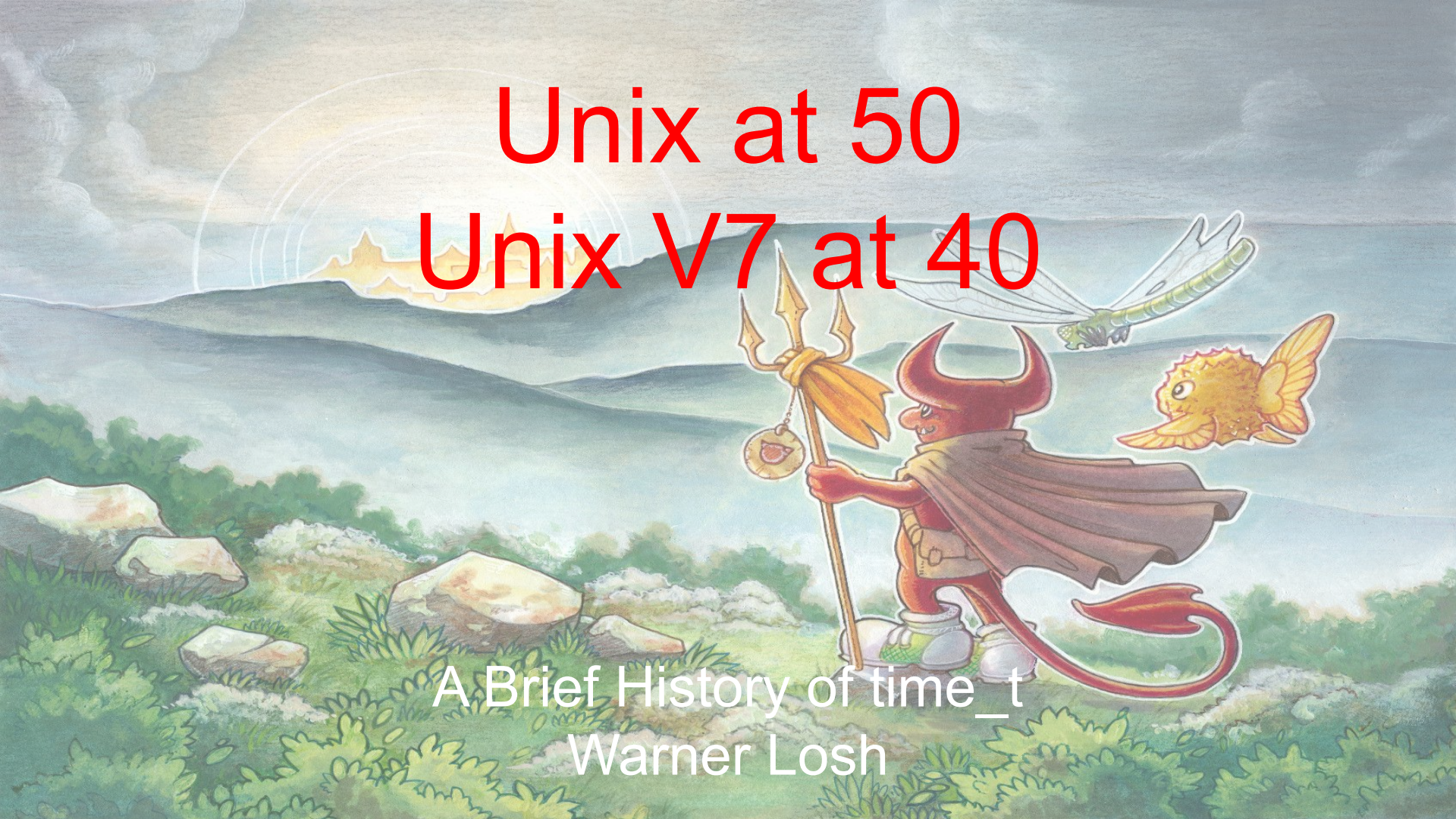
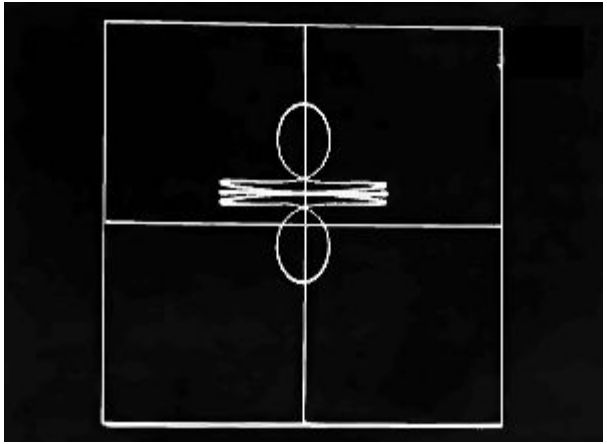
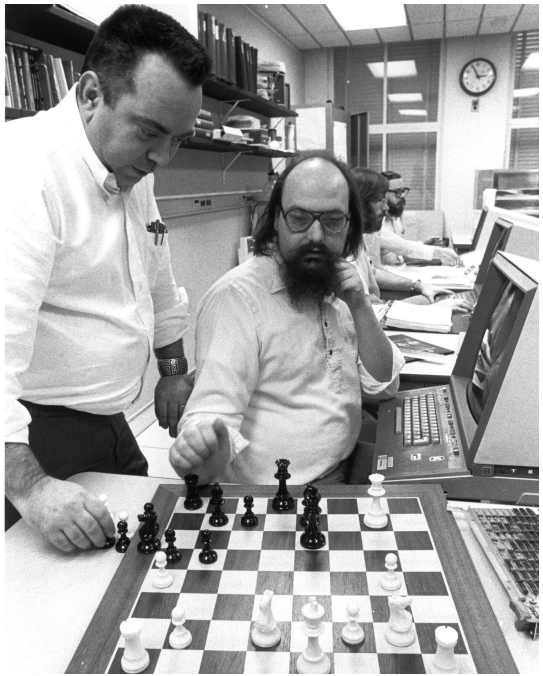
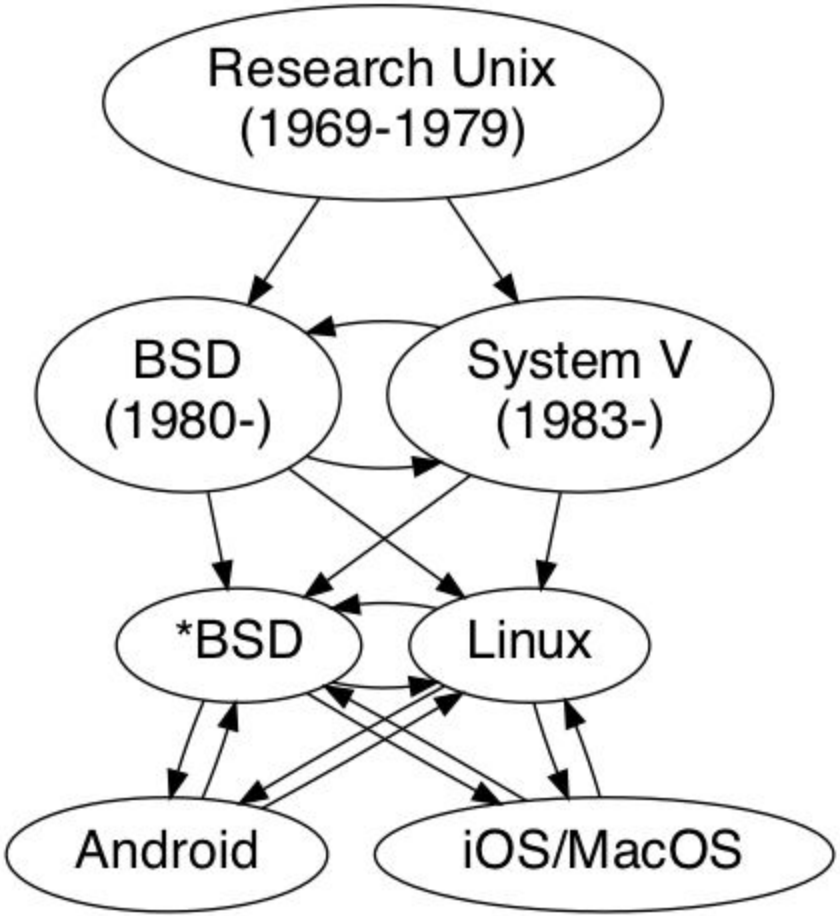

Unix at 50
Unix V7 at 40
A Brief History of time_t
Warner Losh






BELL LABS
SENSITIVE


Research Unix
(1969-1979)
BSD
(1980-)
System V
(1983-)
*BSD
Linux
Android
iOS/MacOS


iPhone

Prehistory: 1950-1969

Bell Labs looked to standardize OS for in-house needs in 1957

BESYS developed for its internal needs, but limited

In 1964 joined Multics effort with MIT and GE rather than continue BESYS

Source: https://en.wikipedia.org/wiki/Bell_Labs#/media/File:Bell_Laboratories_logo.svg

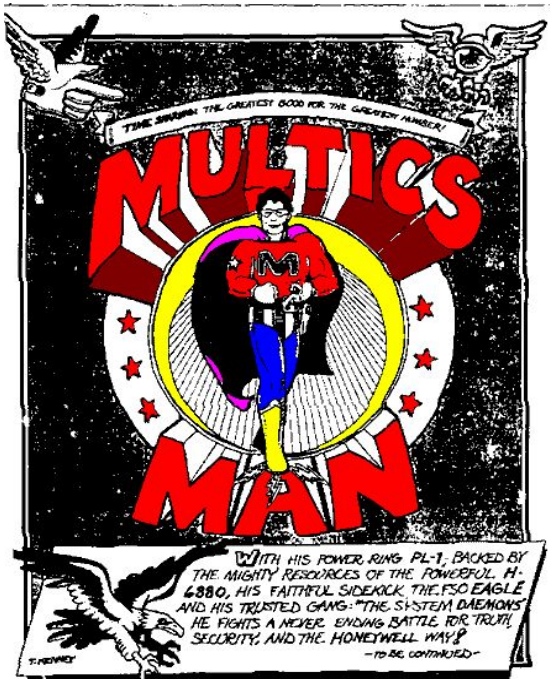
Multics on General Electric / Honeywell GE-645

- 36-bit Mainframe
- Hierarchical File System
- Virtual Memory
- Dynamic Linking
- Implemented in PL/I
- Multiple Languages
- Named Pipes
- ACLs

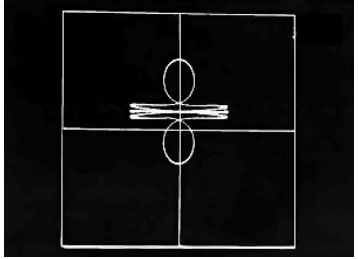
Source: https://multicians.org/drv-bull.html

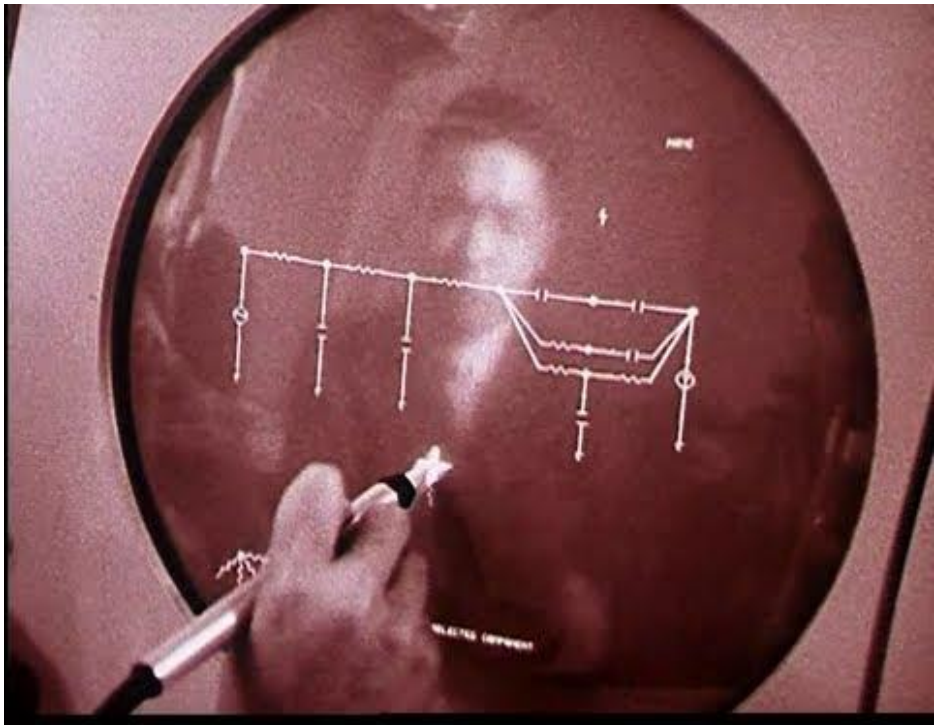
Bell Labs Quit Multics

- Bell Labs Management and Research Scientists frustrated by slow pace
- GE, Bell Labs and MIT each had different goals
- Hard to work together towards common goal
- Bell Labs withdrew from Multics project in 1969
- System too large and complicated
- Required too many resources
- Base hardware too expensive
- Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna last to leave.



Ken’s New System





- Ken loved the game Space Travel
- GECOS version $50-$75 per game
- Ken found discarded PDP-7 from Visual and Acoustics Department
- Ken rewrote GECOS version in PDP-7 assembler
- Transported binaries from GE-635 to PDP-7 on paper tape
- Started Ken’s New System to make Space Travel hacking easier
- Ken started a Fortran compiler, ended up with B...

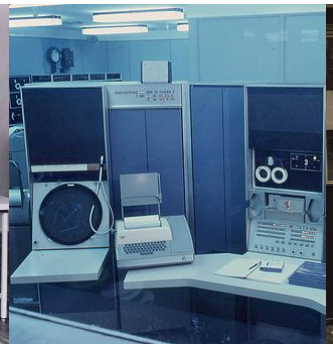
Source: The Incredible Machine 1968 Bell Labs

Wait, what's a PDP-7?

- "Low Cost" 18-bit mini computers from 60s
- Mostly binary compatible from model to model

PDP-1 (59)
~10

PDP-4 (62)
~25



PDP-7 (64)
~100

PDP-9 (66)
~300

PDP-15 (70)
~100

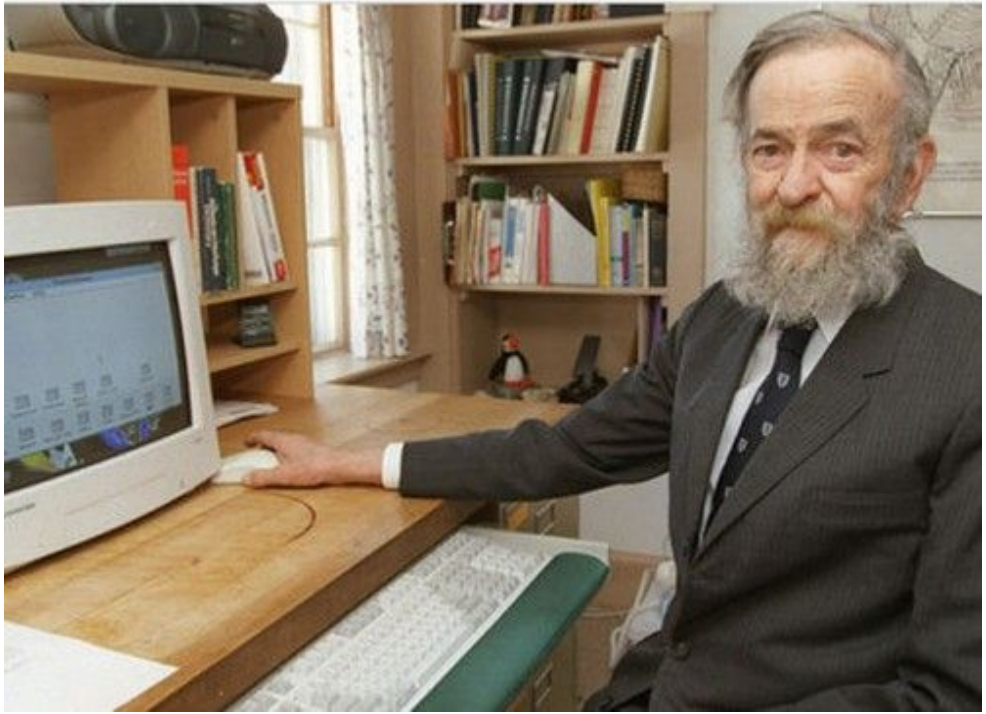

JANUS 71

PDP-7: Ken’s New System becomes Unix

- Thompson implemented “Paper File System” and drivers for paper tape, screen, disk and TTY
- Sources believed lost to time
- Unsure about exact configuration
- Never called Unics
- Others started using pdp7 unix

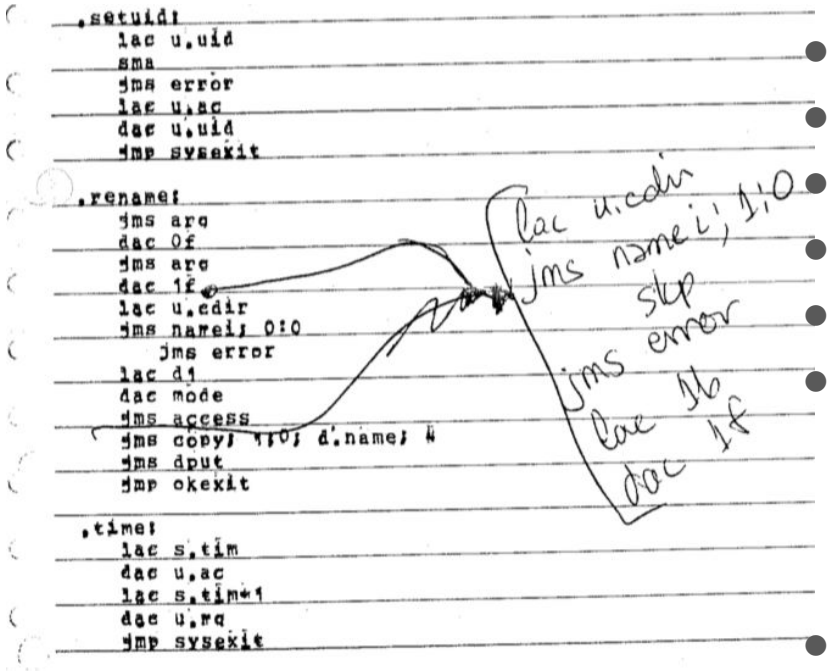
Source: https://www.soemtron.org/images/jpgs/decimages/53077095178cff96d08.jpg

Lost? Not so fast...



- Robert Morris Sr
- Bell Labs from 1960-1986
- Known for crypt, /etc/passwd
- Passed away in 2011
- Had large collection of papers
- Doug McIlroy sorted in 2015
- Found PDP-7 Unix manuals
- Posted on https://tuhs.org/
- Still no source

2016 PDP-7 sources recovered



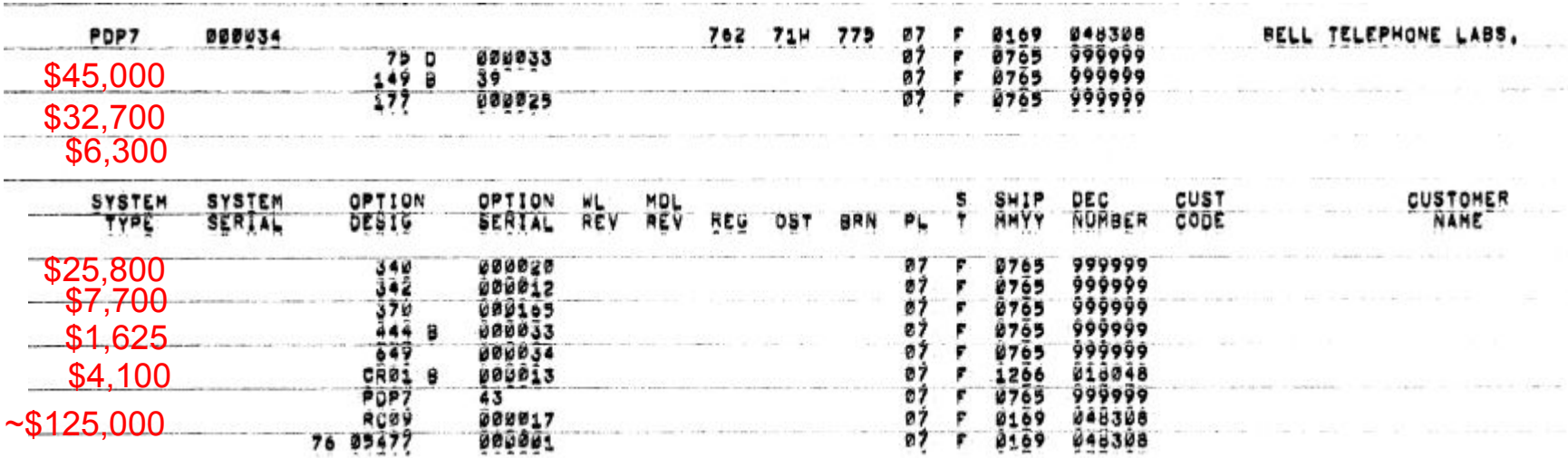
```
.setuid:
    lac u.uid
    sma
    jms error
    lac u.ac
    dac u.uid
    jmp sysexit

.rename:
    jms arg
    dac 0f
    jms arg
    dac 1f
    lac u.cdir
    jms namei; 0:0
        jms error
    lac d1
    dac mode
    jms access
    jms copyi; 1:0; d.name; 4
    jms dput
    jmp okexit

.time:
    lac s.tim
    dac u.ac
    lac s.tim+1
    dac u.mq
    jmp sysexit
```

- Norman Wilson finds PDP-7 sources
- Warren Toomey / TUHS “V0 unix”
- PDP-7 toolchain written
- PDP-7 “user mode” simulator
- SIMH enhanced to run pdp-7 unix
- Userland limited to: adm, als, apr, as, bc, bi, bl, cas, cat, check, chmod, chown, chrm, cp, db, dmabs, ds, dskio, dskres, dsksav, dsw, ed, and mksys
- init, ln, ls, mv and sh newly written
- https://github.com/DoctorWkt/pdp7-unix

pdp7-unix New Discoveries

- Know what devices the kernel supports
- Also found 18 bit field service logs
- Know from simh the disk drive was a retro fit from PDP-9 RB-09
- Know there's tty, light pen, DEC display 340 driver, DISPLAY-2 driver
- No other PDP-7 could run it, none had the disk…
- No other PDP-7 had Bell Labs DISPLAY-2 (though it was option)
- There were 3 PDP-9 and 2 PDP-15 at bell labs that ran pdp7-unix
- IOCCC winner: https://www.ioccc.org/2018/mills/hint.html (pdp-7 emulator to run restored V0)
- Detailed analysis: https://bsdimp.blogspot.com/2019/07/the-pdp-7-where-unix-began.html

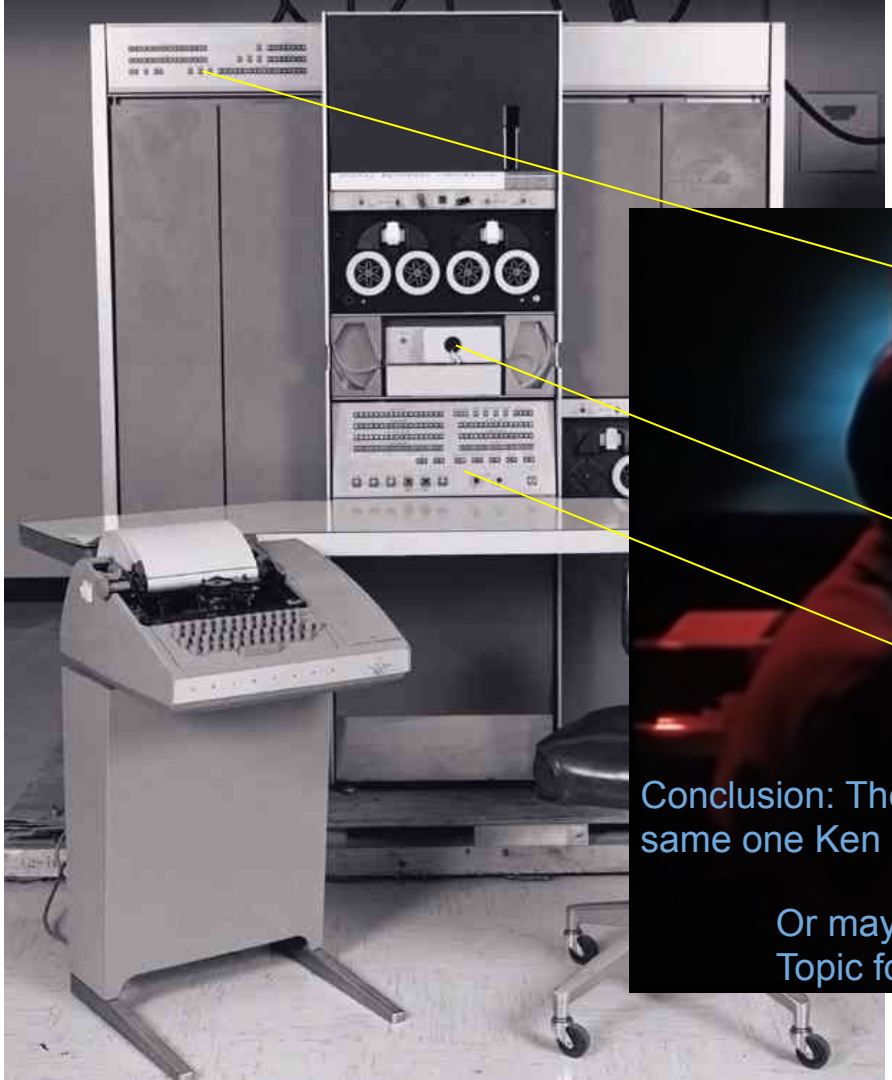
Ken’s Discarded PDP-7 was SN 34

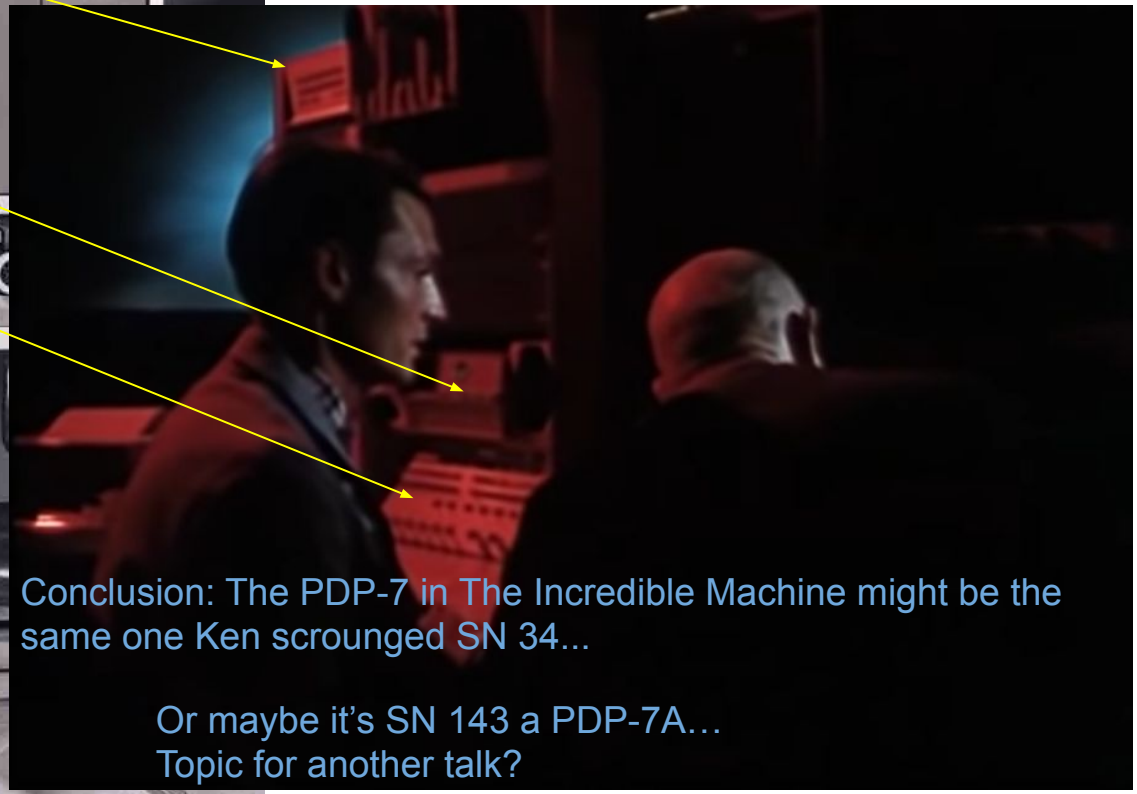


SYSTEM TYPE	SYSTEM SERIAL	OPTION DESIG	OPTION SERIAL	WL REV	MDL REV	REG	OST	BRN	PL	S T	SHIP MMYY	DEC NUMBER	CUST CODE	CUSTOMER NAME
PDP7	000034					762	71W	779	07	F	0169	048308		BELL TELEPHONE LABS,
		75 D	000033						07	F	0765	999999		
$45,000		149 B	39						07	F	0765	999999		
		177	000025						07	F	0765	999999		
$32,700														
$6,300														
$25,800		340	000020						07	F	0765	999999		
$7,700		342	000012						07	F	0765	999999		
$1,625		370	000105						07	F	0765	999999		
$4,100		444 B	000033						07	F	0765	999999		
		647	000034						07	F	0765	999999		
~$125,000		CR01 B	000013						07	F	1266	010048		
		PDP7	43						07	F	0765	999999		
		RC09	000017						07	F	0169	048308		
		76 05477	000001						07	F	0169	048308		

Total: ~$250,000 1965 Price List

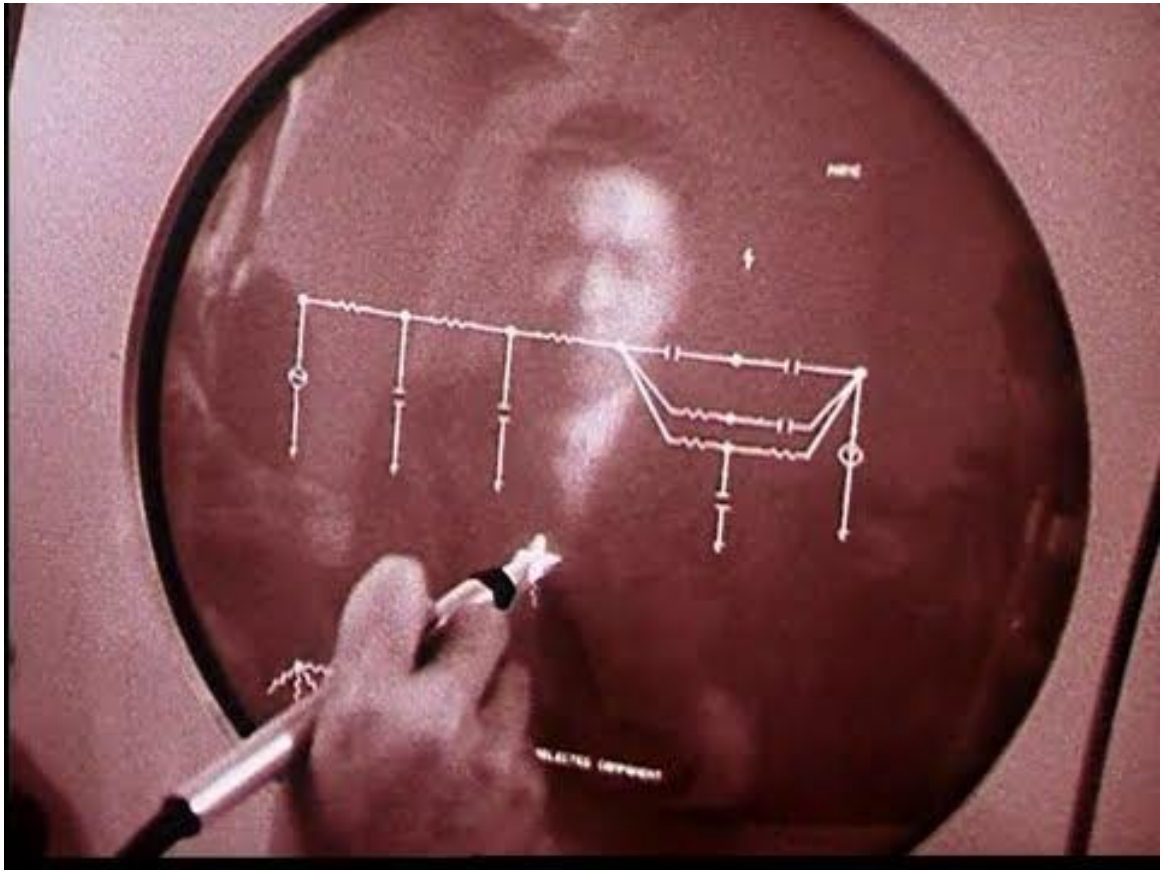
Could it be?





Conclusion: The PDP-7 in The Incredible Machine might be the same one Ken scrounged SN 34...

Or maybe it's SN 143 a PDP-7A…
Topic for another talk?



Transition to PDP-11

- 1970 pdp7 unix's popularity inside Bell Labs Murray Hill helped credibility
- PDP-7 was obsolete even before port began (new it was $250,000)
- PDP-7, PDP-9 and even the new PDP-15 were viewed as having no future
- Popularity justified the purchase of PDP-11/20 for $65,000
- Ken wrote PDP-11 assembler on PDP-7 in B.
- Dennis starts to rewrite B compiler to output PDP-11 code.
- PDP-11/20 delivered to Bell Labs September 1970
- Hard disk so new, it wasn't delivered until December 1970
- No MMU on the 11/20

PDP-11 Port / 1st Edition Unix

- 1st edition rewrote pdp7 unix in PDP-11 assembler (cross built from pdp7)
- Kernel fit into 8kw of PDP-11/20 (DEC CSS builds custom MMU for 11/20)
- Only kernel and init.s sources survive
- B evolves into new language Dennis calls nb or newb (almost to c)
- TUHS has recovered / reconstructed other sources (Unix 1972)
- Binary images available, unclear provenance
  - Multiple “epochs” muddy interpretation of early extant tapes / images
  - Each year new epoch used (60Hz ticks or 2.25 years, v4 shifted to seconds)
- Kernel ~5k lines of assembler
- 1st Edition declared on Nov 3, 1971
- Patent department formatting patent applications one of first users

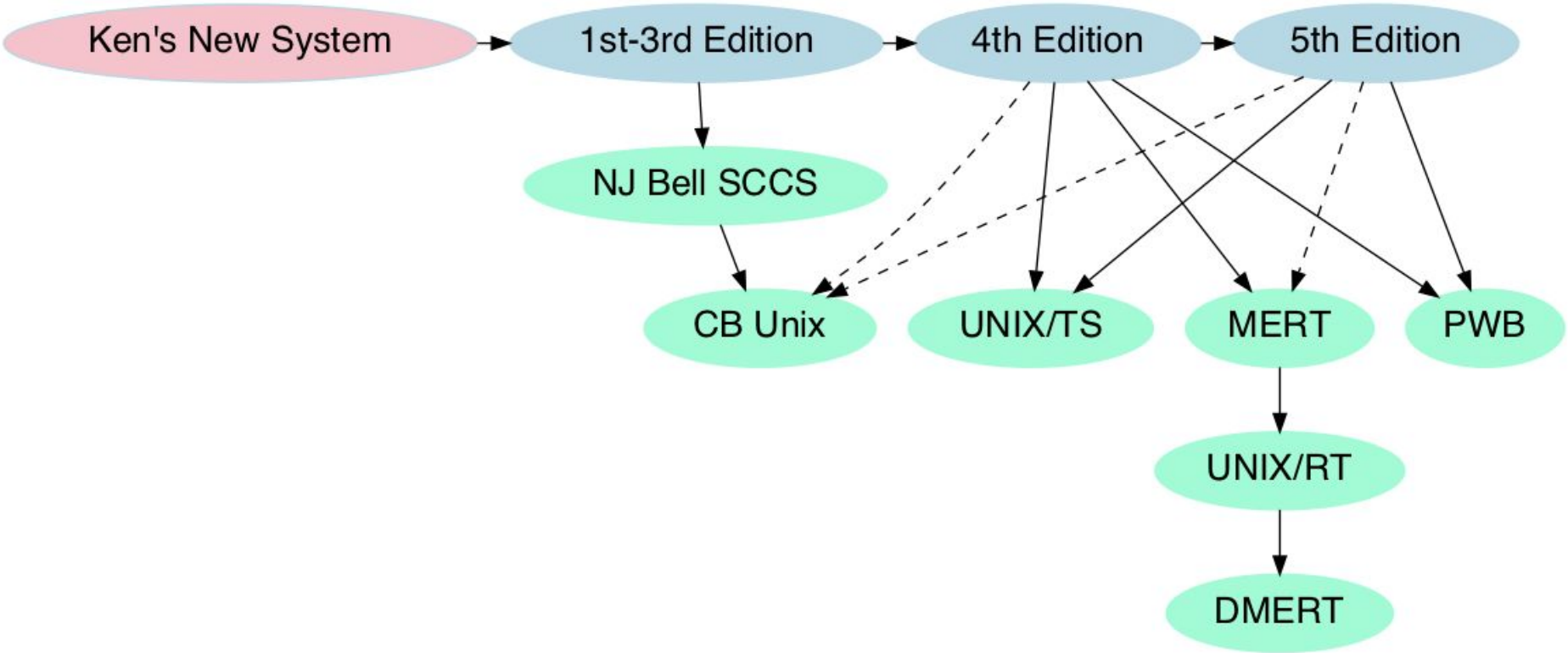
2nd, 3rd, and 4th Editions Unix

- 4th edition first in C, only late version of kernel survives (most userland .s)
- Original Unix paper based on 4th edition (BSTJ 57:6)
- 4th Edition nexus of efforts beyond research (research unix came ‘as is’)
  - BISP 10/73: PWB focused on programming (SCCS and rje for IBMs)
  - OSG 72: CB-Unix turnkey production Unix systems, bug fixes and new kernel features
  - USG 9/73: Berkley Teague Unix/TS with support focused on timesharing
  - 73: MERT (later Unix/RT and DMERT) V4 running under hypervisor
  - First version to be sent to Universities (Columbia and Berkeley, maybe others) Lou Katz
  - Lab’s culture of sharing code one root of today’s open source culture
- Informal Unix Users’ group starts around this time.
- 2nd: June 12, 1972
- 3rd: Feb 1973
- 4th: Nov 1973

Wait, CB-Unix? UNIX/TS? PWB? What's that?

- USG - Unix/TS
  - Research Unix, but with support started. Berkely Tague 1973 (assuming USG did Unix/TS)
- PWB - Programmers Work Bench (BISP)
  - Used by programmers at Bell Labs and released publicly (1973 by Canaday, Mashey, et al)
  - Innovated around making programming easier and more efficient: SCCS, make, rje IBM
  - Ultimately merged with Unix/TS to become System III
- CB-Unix is Columbus Unix
  - Much early innovation adopted by later Unix
  - Merged frequently from Research Unix and PWB
  - Internally distributed, almost all copies lost
  - Group based in Columbus Ohio (based on New Jersey Bell's 1972 Unix port to an ESS confusingly called SCCS that ran on an 11/20)
  - Used by other groups in the company to create products or production systems
  - Ultimately merged with System III to become System V

4th Edition Family Tree



5th edition

- More universities outside of Bell Labs get this
- Still exclusively PDP-11
- Kernel port to C complete
- Userland still mostly in assembler
- First version with complete surviving sources
- PWB / MERT (Unix/RT) / Unix/TS / CB-Unix evolve with imports of Research Unix in the 4th through 6th Edition era (though the exact timing remains murky due to limited extant copies of pre-V6 versions of these branches)
- June 1974

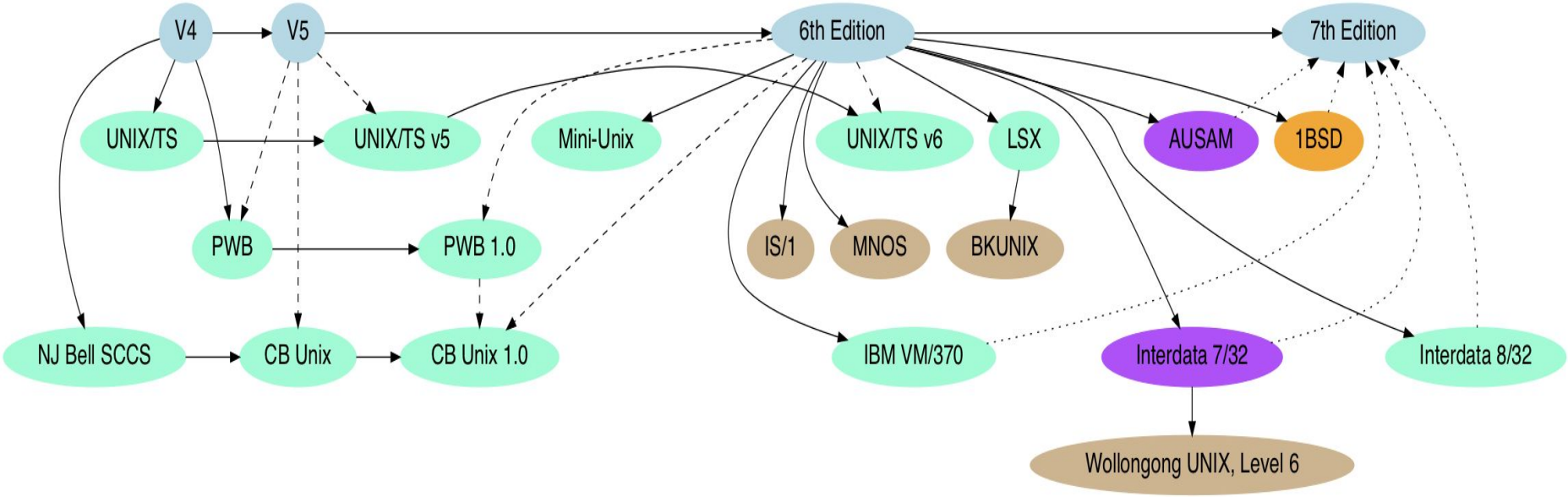
6th Edition May 1975

- Widespread distribution: maybe 100 sites
- Many improvements
  - New system calls
  - Support for newer PDP-11/40 and many new peripherals
  - Mike Lesk's iolib (libp) starts standardizing I/O, but only similar to stdio in v7
  - First tape install system from Dennis Ritchie (RK05 images before)
  - fc -- FORTRAN 66 compiler and sno -- a SNOBOL III compiler
- Beginning of porting efforts
  - Wollongong University ported to Interdata 7/32 (first non-PDP-11 port)
  - Bell Labs ported to VM/360 and Interdata 8/32 (later ports)
- First commercial licenses (RAND had the first one)
- PWB First release (1977) based on v6
- Lions Book

6th Edition May 1975 (cont)

- LSI-UNIX LSX (stripped down version for low end PDP-11)
- Mini-unix was another stripped down version
- AUSAM (U of Sydney and U of New South Wales's V6 distro)
- MNOS (Soviet V6 derived system as IPK Minavtoproma)
- BKUNIX (based on LSX for Elektronika BK (DEC Professional Clone))
- Used at MIT to teach OS class 2002->2006, replaced by xv6 (x86 rewrite)
- CMU and RPI teache using V6
- USENIX incorporates and starts distribution of tapes
- IOCCC winner: https://www.ioccc.org/2018/mills/hint.html (pdp11 emulator on pdp7 emulator)
- RAND Pipes would inspire named pipes later

6th Edition Family Tree



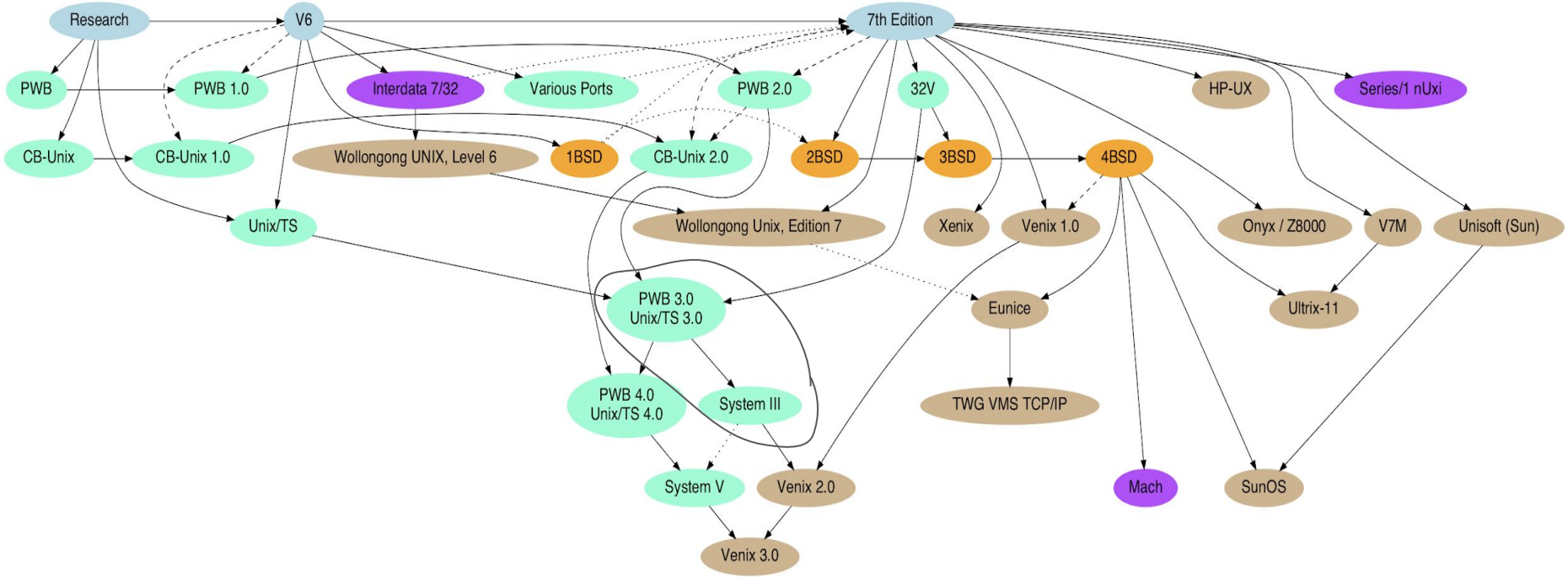
Typesetter C and the “50 changes”

- After 6th Edition
- Newest ‘C’ language features, including
  - long, unsigned and union data types
  - Typedef, struct initializers
  - Bit field support
  - Static support
  - Cpp enhancements
- Dennis Ritchie came up with 50 diffs to take V6 into the then-current Research Unix that were supposedly adopted, at least in part by Unix/TS, PWB and CB-Unix in advance of V7.
- Similar fixes wound up leaking out of Bell Labs… was an early viral spread of data...

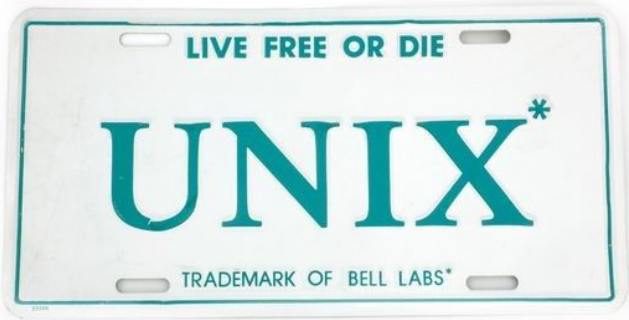
7th Edition Jan 1979 (Best Unix Ever?)

- More Portable
- Custom versions due to new V7 redistribution license.
  - The Wollongong Group (TWG) licenses Wollongong Unix
  - Unisoft ports V7 to 68000 for Standford's SUN Terminal
  - Microsoft ports V7 to many platforms (x86, others) (SCO would do later work)
  - Onyx Systems produce Z8000 with V7 port
  - VAX Port (called 32V)
  - DEC produces V7M which turns into Ultrix-11
  - PWB 2.0 based off of V7
  - VenturaCom produces Venix based on V7
  - Many other companies started with V7 port, but release System III or System V based product
- New syscalls and utilities (more in a bit)

7th Edition Family Tree



Classic License Plates



M

Credit: https://www.mckusick.com/beastie/jpg/foglio.jpg

In V7 or not

- Was head(1) or tail(1) in V7?

In V7 or not

- Was head(1) or tail(1) in V7?
  - tail(1) was in V7. head(1) was invented at Berkeley, since they didn’t know sed 10q :)
- Job Control?

In V7 or not

- Was head(1) or tail(1) in V7?
  - tail(1) was in V7. head(1) was invented at Berkeley, since they didn’t know sed 10q :)
- Job Control?
  - Job control was added by Berkeley by Bill Joy based on work done at MIT by Jim Kulp
- Inherited Environment Variables

In V7 or not

- Was head(1) or tail(1) in V7?
  - tail(1) was in V7. head(1) was invented at Berkeley, since they didn’t know sed 10q :)
- Job Control?
  - Job control was added by Berkeley based on work done at MIT
- Inherited Environment Variables
  - Not in V6, but in V7 with new /bin/sh written by Bourne.
  - One of many cool new features in Bourne’s shell

New Features in V7

- New Development tools: lex, lint and make
- Migrated to portable C compiler (though only with pdp-11 back end)
- Bourne Shell (shell scripts could be in pipeline, v6 sh had script as stdin)
- awk, tar, fortune, at, calendar, and f77
- Networking support (not TCP/IP, but UUCP and datakit)
- New system calls: access, acct, alarm, chroot, exece, ioctl, lseek, umask, utime.
- New stdio, environment variables, getenv, popen and system
- Support for disks and files > 1GB in size…
- New Terminal Interface (yuck)
- Datakit’s mpx IPC method (similar to pipes) that never caught on

V7 Licensing

- Apart from the special Wollongong and ISC licenses, V7 was the first time AT&T had a standard license to allow what we'd call derivative works to be distributed (though at the time it was framed as binary redistribution).
- Compared to later releases, terms were favorable. So many companies based their product on V7, but later rebased it to System III or System V but paid AT&T under the original V7 license. Geared towards minis ($1k/CPU)
- This licensing allowed Unix to grow and flourish outside the labs, but limited its growth in PC space.
- Even so, the license was far from ideal and somewhat confusing
- Full analysis and discussion of this, and subsequent licenses, is outside the scope of this talk.

Venix

- First Unix port commercially available for 8086
  - PC/IX and Xenix were other, more successful ones but with later release dates on 8086
- Combined V7 kernel with userland features from BSD and System III
  - Version 2 has more pulled in from System III, V1 was relatively pure V7 port
  - Version 3 transitioned to System V source, and required MMU / i386+
- Ported to a large range of non-IBM PC 8086 machines
- Was the only Unix game in town for the DEC Rainbow 100

Rainbow 100

- 8086 + Z80 computer that wasn't IBM PC compatible
- Had many cool features for its time (could run both 8-bit CP/M and 16-bit DOS and CP/M 86)
- Largely forgotten by history (see first point)
- Speaker's first computer (which is why I care)
- At the time, I wanted Venix.
  - Couldn't afford it when I was a student.
  - Couldn't find it to buy when I had a real job with money

Credit: https://en.wikipedia.org/wiki/Rainbow_100#/media/File:DEC_Rainbow_100.jpg

Rainbow Venix disks surface

- In September 2016 Bill Degnan purchased a bunch of DEC Rainbow stuff from a former DEC Employee.
- Included was a complete set of not only Venix 86/R disks, but also the even rarer Boston Software Works edition which added support for reasonable winchester drives and had other nice features.
- In April 2017, Bill sent the disks to the speaker to be imaged.
- After many false starts, I imaged the disks and installed Venix 86/R on my Rainbow.
- I then started to wonder: how close was this to Unix V7?

Venix Source Restoration Project

- V7 sources now available
- Disassemble tools
- MAME
- No userland emulation
- Tests on the old Rainbow showed about ½ of the commands from the TUHS V7 archive compiled to an identical binary.
- A full build of V7 on the Rainbow took about 15 hours
- Transfer of results to stable host also takes hours

Venix user emulator

- Executables are a.out format (with STABS debugging)
- 4 flavors of binary
  - NMAGIC (CS and DS = SS = ES)
  - OMAGIC (CS = DS = SS = ES)
  - OMAGIC and NMAGIC with stack size
- 64k address space (or semi-split I&D space 64k each)
- X86 instruction emulator
- Venix system calls are introduced with “int 0xf1”
- Exec loader to start image
- All binaries static (no shared libraries)

X86 Emulators

- Chose one I found that was simple and targetted at IBM PC emulation
- Tried QEMU user mode stuff, but the 16-bit execution environment would have been a lot of work
- X11 comes with one to execute BIOS code in video cards (including 16-bit code)
- MAME, pcemul, and most of the DOS emulators have instruction emulators these days
- Create syscall table in FreeBSD (16-bit mode won’t work on amd64)

Bringing the pieces together

- Reverse engineering the system call interface
  - Then finding it documented in the Venix/11 manuals which had call outs for 8086
- Adding all the translation code
  - Writing helper functions to fetch DS:AX strings and the like
  - I had forgotten how much 16-bit 8086 code bugged me
- Test, test, test
  - At first only with the Venix binaries I had (my Rainbow had issues for a while)
  - Then with custom binaries to test edge cases (once the Rainbow was back)

Loading binaries

OMAGIC

Tiny Model

CS = DS = ES = SS = load point

Load text, skip stack, load data, zero BSS

Setup segment registers and IP

Copy args and env into stack as args to entry point

Hand off to x86 emulator

Section
Text
Stack
Data
BSS

Loading binaries

NMAGIC

Small Model

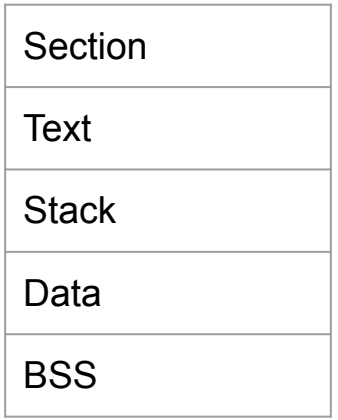
CS = load point, DS = ES = SS = (CS + text/16)

Load text, skip stack, load data, zero BSS

Setup segment registers and IP

Copy args and env into stack as args to entry point

Hand off to x86 emulator



Example system call -- open(2)

```
/ C library -- open

/ file = open(string, mode)
/ file == -1 means error

.globl  _open, _errno

_open:
        push    bp
        mov     bp,sp
        mov     bx,#5
        mov     ax,*4(bp)       | string
        mov     dx,*6(bp)       | mode
        int     $f1
        jcxz    1f
        mov     _errno,cx
1:
        pop     bp
        ret
```

- ds:ax has file to open
- dx has mode
- On return, ax has fd
- Kernel returns success/failure in C bit
- Errno return in cx

- Kernel has to try to open file natively, and if it succeeds assign a slot in its file stable and return an index to that slot, or -1 and an error in CX. Note: on error, ax will be 0xffff. The kernel only signals with C bit to write to libc's copy of errno

Current Status

- All the basics work (open, read, write, close, ioctl)
- Fork / exec do not (I need to restructure what I've written)
- About ¾ of the commands work or mostly work
- Commands that know the filesystem structure (df, fcheck) don't
- Assembler working, c compiler not
- Awk works for some things
- The shell and make have big issues
- Ported FreeBSD's ddb for tracebacks, etc
- All roads lead back to fork/exec
- Looking at better x86 emulator to pick up floating point

Future Work

- Refactor loader so it can be used for exec (I should have written it like that in the first place)
- Improve debugger to find out what’s going wrong
- Investigate other x86 emulators (there’s a lot). The one I chose was a bit too simple.
- Refactor process state into an array of proc structures so I can fork / exec, but still share memory space.
- Make more use of MAME instead of DEC Rainbow now MAME Rainbow emulation is better and my poor old DEC Rainbow is showing its age.

Special Thanks

- Clem Cole who provided much remembrance to help me get details right…
- Thanks to the TUHS mailing list who commented on my family tree diagrams and details in this talk and pushed me to pedantically document where “living history memory” differed from the historic record...
- https://www.tuhs.org has the early Unix trees and manuals that let me confirm / deny people’s memories of the time. Such efforts are ongoing...
- Thanks to the following folks that provided feedback on these slides: Rob Pike, Dan Cross, Warren Toomey, Arnold, Branden Robinson, Larry McVoy, Jon Steinhart, Diomidis Spinellis, Wesley Parish, Bakul Shah, William Petcher, and Theodore Ts’o.

Questions

Warner Losh
imp@bsdimp.com

Marked Safe From

The Unix Wars

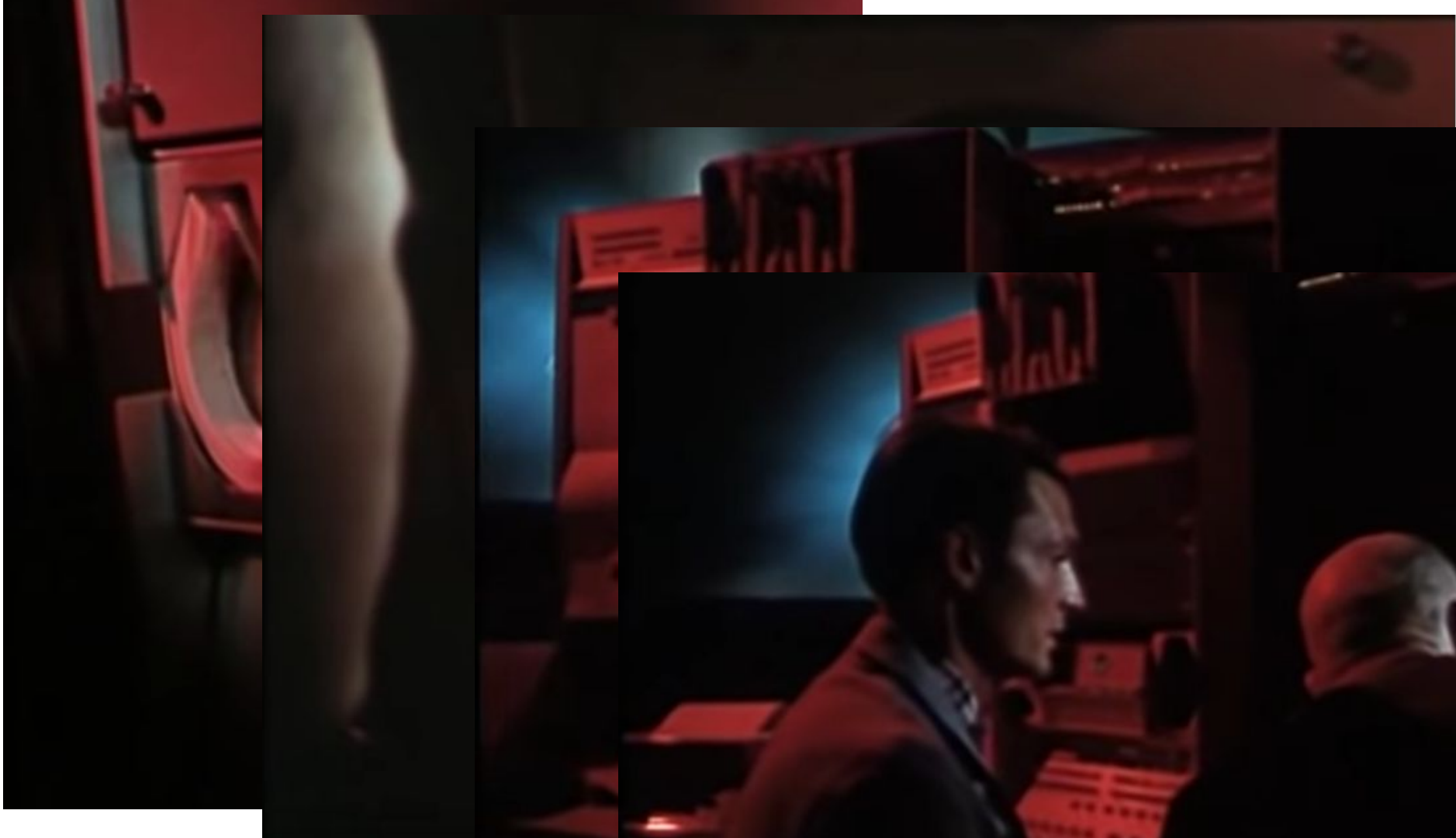
Today

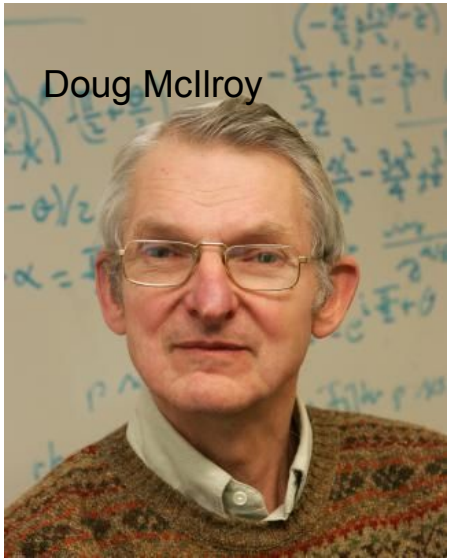
http://people.freebsd.org/~imp/talks/eurobsdcon-slides-2019.pdf


R.G. STUART

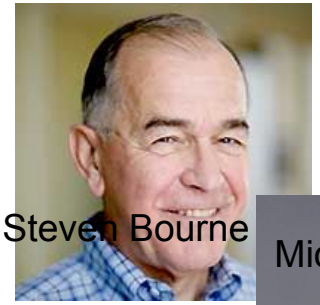
- https://www.youtube.com/watch?v=iwVu2BWLZqA
- https://www.youtube.com/watch?v=tc4ROCJYbm0
- https://www.youtube.com/watch?v=XvDZLjaCJuw

Good archival footage

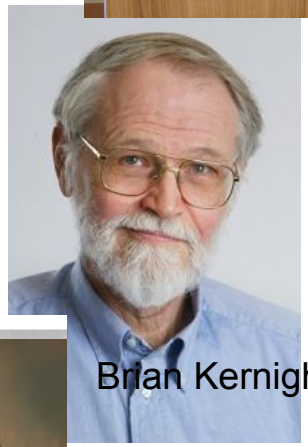



Doug McIlroy

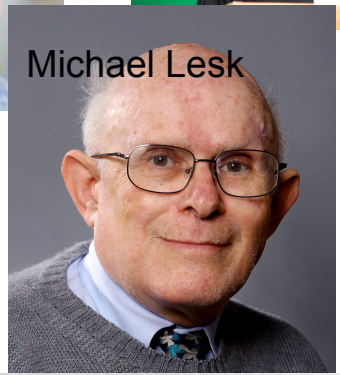

Peter
Neumann


Steven Bourne


Rob Pike


Brian Kernigh


David Korn
KORN
KORN
REQUEST


Michael Lesk


Rudd Canaday


Gordon Bell


Joe Ossanna
(nroff / troff)