History of the Shell

In the beginning was paper

Presentation shared with the

Saint Louis Unix Users Group

2024-01-10

By Ed Howland < @edhowland@fosstodon.org >

Welcome

My Time Capsule

Advancing thru the ages:

- From HP Desk Calculator to BASIC login over 300 baud A/C modem to remote mini.
- Sys Op on dual Data General Nova minis with Teletype consoles and front panel toggles.
- Trained on boot operation of I.M.P. Honeywell 515 for our local node of the Internet.
- As a **supplicant**, did punch card programming in COBOL, later FORTRAN, then PL/1.
- “Promoted” to **acolite** and allowed to actually login to IBM mainframe on glass TTY.

History Builds

0. Punch card/paper tape batch data processing.
   - ***Resident Monitors ~ simple terminate and stay resident style programs.***
1. Simple screen consoles.
   - IPL (Initial Program Load)
   - Diagnostics and monitoring.
2. Multi-processing and simultaneous jobs.
   - Divide and conquer because I/O is slower than CPU.
   - So, work inside the blocked wait cycles.
3. JCL (Job Control Language)
   - Later versions have features similar to shells developed in the 70s & 80s.

Time Gets Going

Things start to move

- Arrival of user terminals where entire Universes are born with Big Bangs and die in heat deaths between the interval of key presses of fast typists.

U. S. Navy and M.I.T. Project Whirlwind

- Attempt to create a flight simulator (in 1944!) .

Time Sharing

Dartmouth Time Sharing System (DTSS)

- It was the first successful large-scale time-sharing system to be implemented (1963-1964) .
- Had user file areas and simple commands to run programs.
- Dartmouth B.A.S.I.C. was actually part of Dartmouth Time Sharing which had user file areas and simple commands to run programs.

Time Moves Faster

Start of the CTSS.

- Fernando Corvabo proposed time slicing between user requests
- Used user passwords to prevent file clobbering and ring based security

From there to MULTICS.

- Hardware: General Electric (645 – delivered late)
    - GECOS (1962) starting off point on G.E. mainframes.
    - First example of 2nd generation of OSes.
    - Version 3 used time sharing
- Software: M.I.T. (OS and PL/1 High level PL to program it in)
- Communications: AT&T

CICS Terminal Server

Originally designed to service utility companies, starting with Michigan Bell (1966).

- Supported these terminals
    - IBM Selectric typewriter (1965)
    - IBM 2260 VDU (1967)
    - IBM 3270 (1972)
- Incorporated many fine ideas that became ubiquitous later:
    - Efficient use of memory and I/O
    - Multi-threaded
    - A sort of open source style software sharing ecosystem.
        - Many ideas incorporated in later web servers
    - Apps written for CICS included dashboard style operator's panels
        - Simple command line processor for maintenance commands

Simple Shells Arrive

Simple shells like Thompson shell and PWB.

- Called V6, the Thompson shell supported redirection operators and pipes.
- Doug McElroy came up with the concept of pipeline.
- Only simple if statements and goto statements.
- Support for background jobs with ‘&' (ampersand) .
- Globing was only done with an external program.
- No shell variables until the Programmer’s Work Bench or PWB shell.

Programmers Work Bench

The PWB shell

- PWB firsts: SCCS, troff/nroff, the find command, xargs, egrep, fgrep, and yacc/lex.
- The PWB shell introduced if/then/else, while statements for looping, the switch statement and shell variables (only 1 letter variable names) .
- PWB introduce the PATH directory list for other binaries besides those in /bin and /usr/bin which users could not write to.
- PWB could use string interpolation for variables.

Steve Bourne Influence

Arrival of Steve Bourne in 1975 from Cambridge, U.K.

- Was a big fan of ALGOL/68 where we get much of Bourne Shell syntax.
- And the rest is history

Then a long progression of incremental improvements.

- Like Korn shell, C shell and then Bash.
- Which is not going anywhere soon.

Future History of the Shell

The next leap?

- Forward might be Powershell with its addition of object pipelines alongside of simple text only streams.

Nushell (~2020) begs, borrows and steals from all that has gone before.

- Focusing on running structured data through data pipelines.
- It also borrows ideas from SQL and relational database techniques.

Feedback

What are your questions?

Presentation shared with the

Saint Louis Unix Users Group

2024-01-10

By Ed Howland < @edhowland@fosstodon.org >

Questions