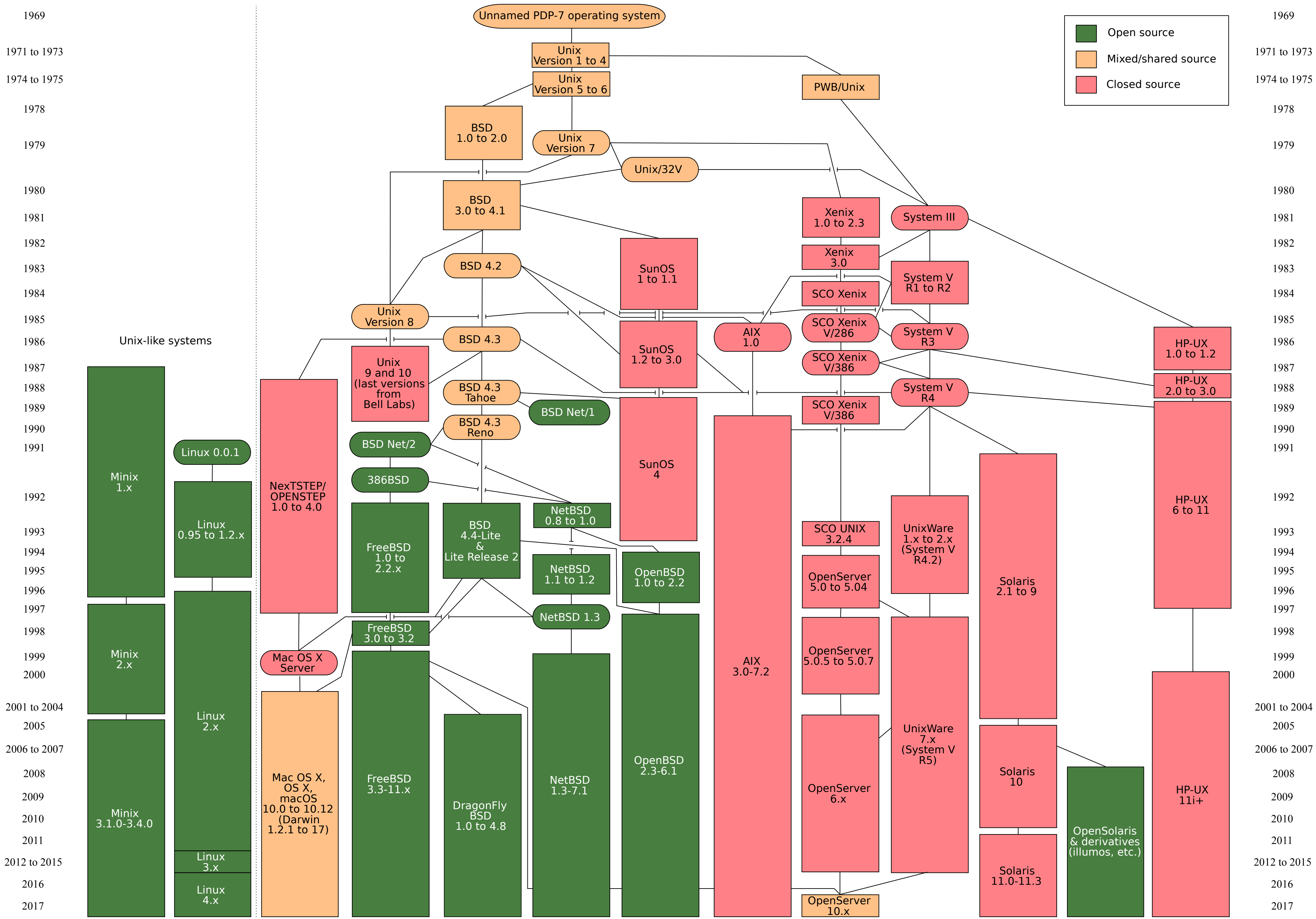
Introduction to the UNIX Environment: Computing as Praxis

Claude Rubinson, Ph.D.
University of Houston—Downtown
rubinsonc@uhd.edu
cjr@grundrisse.org

ACM UHD
Houston, Texas
October 29, 2019

Why does a 50 year-old technology dominate modern computing?


🟩 Open source
🟧 Mixed/shared source
🟥 Closed source
1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2007
2008
2009
2010
2011
2012 to 2015
2016
2017
Unix-like systems
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
SunOS 1 to 1.1
BSD 4.2
System V R1 to R2
SCO Xenix
Unix Version 8
AIX 1.0
SCO Xenix V/286
System V R3
BSD 4.3
SunOS 1.2 to 3.0
HP-UX 1.0 to 1.2
Unix 9 and 10 (last versions from Bell Labs)
SCO Xenix V/386
HP-UX 2.0 to 3.0
BSD 4.3 Tahoe
System V R4
BSD Net/1
SCO Xenix V/386
BSD 4.3 Reno
BSD Net/2
Linux 0.0.1
Minix 1.x
NexTSTEP/ OPENSTEP 1.0 to 4.0
SunOS 4
386BSD
HP-UX 6 to 11
Linux 0.95 to 1.2.x
FreeBSD 1.0 to 2.2.x
BSD 4.4-Lite & Lite Release 2
NetBSD 0.8 to 1.0
SCO UNIX 3.2.4
UnixWare 1.x to 2.x (System V R4.2)
Solaris 2.1 to 9
NetBSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.04
NetBSD 1.3
FreeBSD 3.0 to 3.2
Minix 2.x
Linux 2.x
Mac OS X Server
AIX 3.0-7.2
OpenServer 5.0.5 to 5.0.7
HP-UX 11i+
Mac OS X, OS X, macOS 10.0 to 10.12 (Darwin 1.2.1 to 17)
FreeBSD 3.3-11.x
DragonFly BSD 1.0 to 4.8
NetBSD 1.3-7.1
OpenBSD 2.3-6.1
OpenServer 6.x
UnixWare 7.x (System V R5)
Solaris 10
OpenSolaris & derivatives (illumos, etc.)
Minix 3.1.0-3.4.0
Solaris 11.0-11.3
Linux 3.x
Linux 4.x
OpenServer 10.x

Resources

- Mike Gancarz, *The UNIX Philosophy* (1995, Digital Press)
- Eric Raymond*, The Art of UNIX Programming* (2004, Addison-Wesley)
- Thomas Scoville, “UNIX as Literature” (1998, Miller Freeman)
- Richard Gabriel, “The Rise of Worse is Better” (1989)

The UNIX Philosophy

- Assume that users are intelligent and know what they’re doing.
  - “UNIX gives you just enough rope to hang yourself—and then a couple of more feet, just to be sure.”
  - “UNIX was not designed to stop you from doing stupid things, because that would also stop you from doing clever things.”
- Complexity is dangerous:
  privilege simplicity above correctness, consistency, and completeness.
- Nobody can predict the future; be humble and remain flexible.

The UNIX Environment

- UNIX is not merely a platform upon which to run applications; rather, the commandline is your *primary* interface.
- A properly designed UNIX program, shell script, or shell function will *integrate* into your environment; can think of UNIX programs as “extensions” or “plugins” to the environment.
- Over the long-term, you customize your environment to your needs, preferences, and style.

Gancarz’s Tenets of UNIX

1. Small is beautiful
2. Make each program do one thing well.
3. Build a prototype as soon as possible.
4. Choose portability over efficiency
5. Store [numerical] data in flat ASCII files.
6. Use software leverage to your advantage.
7. Use shell scripts to increase leverage and portability.
8. Avoid captive user interfaces.
9. Make every program a filter.

Small is beautiful; Make each program do one thing well

Small programs:

- are easy to understand.
- are easy to maintain.
- consume fewer system resources.
- are easier to combine with other tools.

Make every program a filter; Avoid captive user interfaces

- All computer programs *are* filters; UNIX makes this explicit and puts the user in control.
  - cf., GUI applications and racist machine learning algorithms
- CUIs trap data and assume that their user is human, which means that they don’t interface well with other programs.

Build a prototype as soon as possible

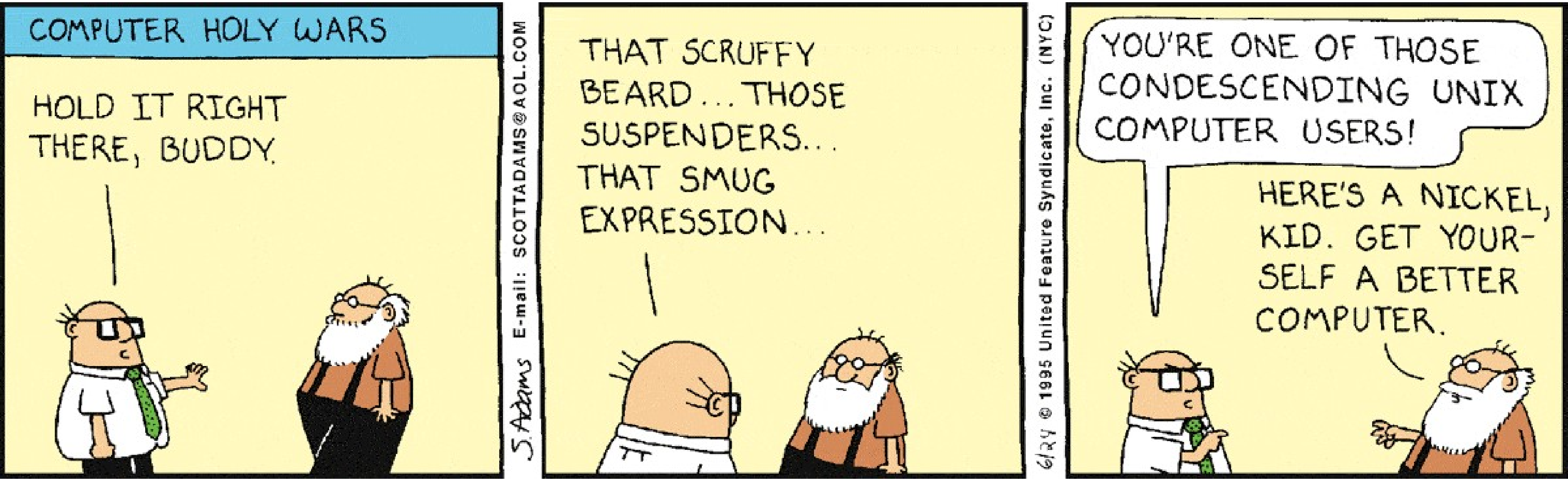
- Shell scripting encourages rapid, iterative test/rewrite development
  - cf., *Manifesto for Agile Software Development* (2001)
- “Worse is better”
  - Seek the 90% solution
  - “When in doubt, use brute force.”

Choose portability over efficiency; Store [numerical] data in flat ASCII files

- The UNIX environment generally assumes plain text and, therefore, provides a rich set of *line-oriented* text manipulation tools (e.g., editors and version control) and languages (e.g., grep/sed/awk, sort|uniq -c, calc/bc/dc).
- stdin, stdout, and stderr don’t distinguish between plain text and binary streams.

Use software leverage to your advantage; Use shell scripts to increase leverage and portability

- Shell scripts allow you to build up your analysis incrementally by piping from stdout to stdin.
- Other programs are treated as black boxes.
- Is often easier to call other programs from the shell than to shell out.
- Shell pipelines are often highly performant
- But: largely unidirectional (FIFOs and temp files can help here).


COMPUTER HOLY WARS
HOLD IT RIGHT THERE, BUDDY.
THAT SCRUFFY BEARD... THOSE SUSPENDERS... THAT SMUG EXPRESSION...
YOU'RE ONE OF THOSE CONDESCENDING UNIX COMPUTER USERS!
HERE'S A NICKEL, KID. GET YOURSELF A BETTER COMPUTER.
S. Adams
E-mail: SCOTTADAMS@AOL.COM
6/24 © 1995 United Feature Syndicate, Inc. (NYC)

"UNIX is user-friendly. It's just picky about who its friends are."