

The origin and significance of the UNIX pipe

UUASC OC

2009-10-12

Dave Close

Precursors

- Communication files: Dartmouth Time-sharing System
- IO redirection: OS/360 JCL and others
- Macros: "switchyard for data streams" - Doug McIlroy

Evolution

- **Prefix notation:** `cat (grep (who))`
- **Doug McIlroy:** `who > grep > cat`
- **Brian Kernighan:** `who ^ grep ^ cat`
- **Ken Thompson:** `who | grep | cat`

Evolution

- **Prefix notation:** `cat (grep (who))`
- **Doug McIlroy:** `who > grep > cat`
- **Brian Kernighan:** `who ^ grep ^ cat`
- **Ken Thompson:** `who | grep | cat`
- `---`
- **BASH:** `cat $(grep $(who))`

The toolbox

- Q: "Unix looked different after pipes?"
- Doug McIlroy: "Yes, the philosophy that everybody started putting forth, 'This is the Unix philosophy. Write programs that do one thing and do it well. Write programs to work together. Write programs that handle text streams, because that is a universal interface.' All of those ideas, which add up to the tool approach, might have been there in some unformed way prior to pipes, but they really came in afterwards."

Impact

- The toolbox approach
- "Little languages": `awk`, `eqn`, `tbl`, `make`
- Named pipes
- Network pipes: `nc`, `socat`
- Sockets

Limitations

- Unidirectional
- Common ancestor
- One-to-one, no multicast
- One input and one output per process

References

- *A Quarter Century of UNIX*,
Peter H. Salus, Addison-Wesley, 1994, 0-201-54777-5
- *Unix: An Oral History*,
<http://www.princeton.edu/~hos/frs122/unixhist/finalhis.htm>
- *Unix Pipes -- powerful and elegant programming paradigm*,
<http://www.softpanorama.org/Scripting/pipes.shtml>