Impact of UCSD Pascal
A View from the Trenches

Roger Sumner

UCSD

Before The Flood

- Languages of the day: Fortran, COBOL, PL/I, RPG(!)
- Batch Mainframe Computers
  - Punch cards, line printers, reel-to-reel tapes, overnight batch runs
  - HUGELY EXPENSIVE
- Minicomputers: Data General, DEC PDP-10, PDP-11, early Prime

Rise of the Microcomputer

- Assembly language
- Basic (many dialects)
- PL/M (Digital Research, Intel)
- All limited to specific platforms
- Very difficult to use
- Open need: portable, easy to use, powerful language + compiler + system

Pascal As A New Language

- Other academic/niche languages: Simula, Euler, APL, Lisp, Algol; all on mainframes
- Pascal was an academic language, too, but…
  - Simple enough for a modest-sized compiler
  - Powerful enough to write a compiler!
  - Pointers as a full-fledged data type
  - Enabled recursive data structures

Key Features of Pascal

- Strong type checking - parameters & expressions
- Pointer types
  - Safe, coupled to specific types (unlike "C" at the time)
  - Allows easily expressed recursive data structures
- Very easy to like!
  - Structured programming wasn't common in those days
  - UCSD Pascal popularized Pascal itself, which was then picked up by…
    - Microsoft Pascal
    - Borland Turbo Pascal
    - Apple Pascal (32-bit)
  - Adopted as a standard development language by many organizations, but…

Limitations!

- What Brian Kernighan saw as missing:
  - Strings, variable size arrays, logical ops
  - Separate compilation, libraries
  - Flexible I/O
  - An "escape" from compiler builtins & dogma
- Lots of folks modified Pascal to solve these issues
  - Raising the ire of the theologians
  - But you gotta get your work done!

In the meantime… The revenge of The Phone Company (TPC)

- C started its march towards domination
  - Low-level strings (OK with libraries)
  - Very flexible separate compilation (dangers, too!)
  - Pointers to functions
  - Very flexible I/O mechanism
  - Pointer/array semantics more powerful, but more dangerous
  - Lear Jet versus Piper Cub? (not in 1974!)

Pascal is dead, its Legacy lives on

- Pascal has gone the way of Latin
- Borland Delphi = Pascal++ without credit
- Key concepts live on, especially Java VM
  - Portable virtual machine
  - Friendlier, safer language than "C"
- Strong type checking, scientific approach to program architecture