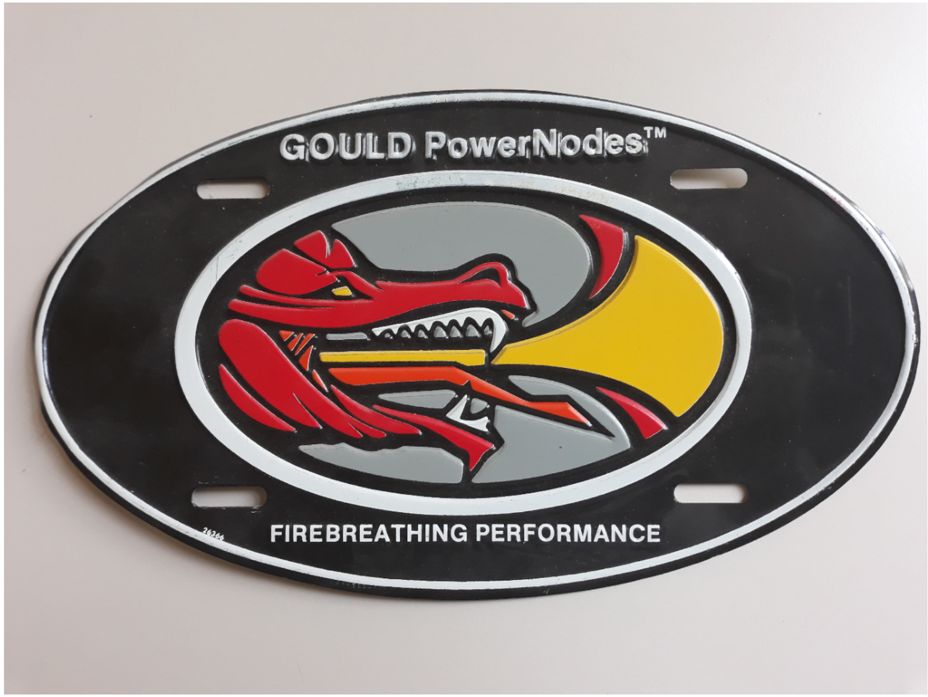
UTX/32 and the Art of Fire Breathing

a long forgotten chapter in UNIX history

UTX - Universal Timesharing Executive

UTX/32 is an operating system that dates back to the 1980s and was designed for Gould/SEL 32-bit machines. Competing with DEC VAX 11/780, the Gould PowerNode 6000 and 9000 systems were very powerful and cost effective. Marketing labeled them fire-breathing dragons.



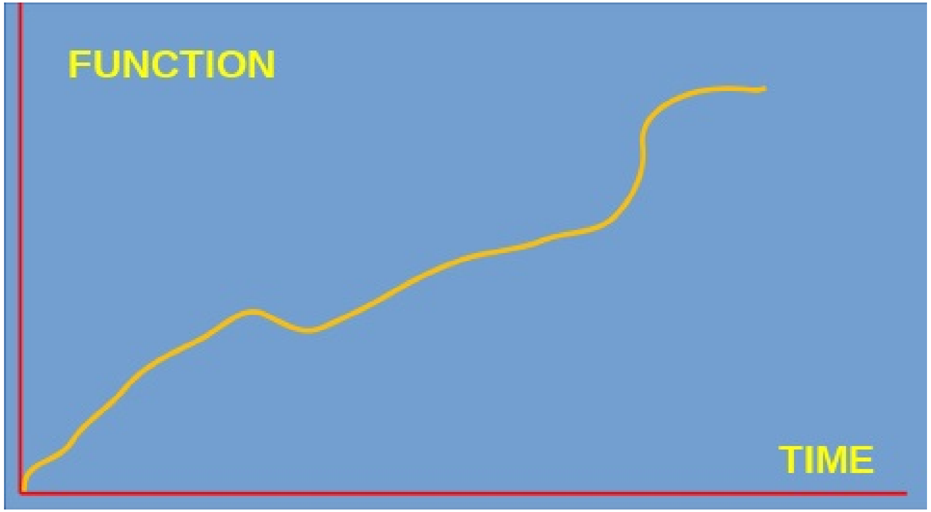
My name is Geert Rolf.

- First contact with UNIX in 1977.
- Hogere Informatica Opleiding 1982.
- Owner / gatekeeper of “Tehuis voor Bejaarde Computers”. (closed 2022)
- Obtained Gould PowerNode 6040 in 1994.
  - Including spare parts, software and hardware docs, diagnostics, distribution tapes for OS + UNIFY rdbms.
  - Mainly used to read magtapes: robust, high speed tape streaming.
  - Last 4 years: sorting out puzzles in SIMH/sel32 project.

What about some history?

- UNIX developments in early 80s.
- Transition of 16 to 32 bit computers..
- A lot of Micro computers on the market.
- The UNIX war: AT&T System III, V versus BSD 4.x technology.
- Mix of developments in UTX/32: Berkeley UNIX, AT&T System V, SUN addons and Real time extensions for UNIX.
- SIMH project: simulation of long gone computers.
- This talk:
  - LESS *need* to know.
  - MORE *fun* to know.

l'histoire se répète... growing up takes time!



Development Curve.

- First **post WWII computers** could only run single job*.* Multitasking and time-sharing came years later.
- Same play with **mini computers**. At first booted from paper tapes.
- *No exception for UNIX*. At first booted from paper tapes.

Ken Thompson must have known...

- ...that 16 bit word size wasn't big enough in the end.
- Ken et al proposed a PDP-10, a 36 bit machine ~ $500K. Management rejected project: "*we don't do operating systems anymore*".
- Bell Labs patent department needed specific type-setting tools.
- 1970: first, small PDP-11 ~ $50K
- **PDP-11 did byte addressing!**

UNIX conquered the world on PDP-11s

- The limit of 64KB triggered a brilliant design: a pipe with a **chain of functions** to get your job done & /bin/sh as **user program**.
- First production UNIX on a PDP-11/20 did **not use mapping / protection**.
- Cheaper PDP-11 **boosted** UNIX into academic world.

In the meantime at U.C. Berkeley

- Computer Systems Research Group (CSRG) offered Berkeley Software Distribution for PDP-11 and later the VAX.
- DoD DARPA contract:
  - DEC VAX preferred platform.
  - OS either VAX/VMS or UNIX.

  UNIX won after 2nd run of benchmarks & on portability to new platforms in the future.

- Project focus:
  - auto reboot,
  - improved signals,
  - socket etc, select,
  - IP,
  - fast filesystem,
  - improved VM.
- BSD 2.11 … 4.4 … BSDi … FreeBSD etc.
- Mt Xinu: commercially licensed BSD initially for VAX.

UNIX Micro Catalog (EUUG; 1983)

~120 UNIX systems.

- proprietary processors still present.
- 85x Motorola 68000 / 68010
- Still 4x Intel 8086
- 12x Zilog Z8000
- 2x NatSemi 16032
- 46x Multibus; 9x VME bus.
- 16MB accepted memory limit

- 57x UNIX v7 port
- 43x System III
- ~10x BSD or BSD compatible
- Awaiting:
  - i386 in 1985...
  - multiprocessor designs to gain processing power...

SYSTEMS ENGINEERING LABORATORIES

- Systems Engineering Laboratories (SEL) **founded in 1961** and based in Fort Lauderdale, Florida.
- SEL designed and built minicomputers **focussed on real-time** applications.
- A range of computers, the SEL 800 series, was introduced.
- Mid 60s: SEL 840 used for telemetry / data acquisition at NASA to troubleshoot Rocketdyne F1 engines for the first stage of the Saturn V moon rocket.
- By 1975
  - SEL launched a multitasking **32-bit system:** the SEL-32/55 for realtime market.
  - O.S. Mapped Programming Executive, **MPX/32, released.**

1981: Gould Electronics takes over

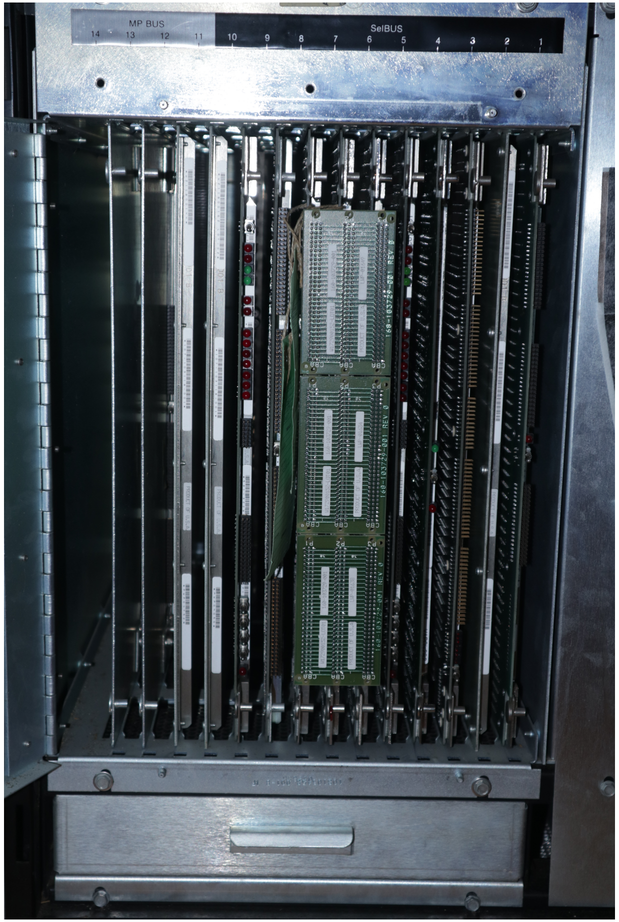
- In 1981 SEL was **taken over by Gould Electronics** and renamed Gould Computer Systems Division (Gould CSD).
- The SEL-32 series was rebranded **Concept-32**.
- In 1988 Gould Electronics was acquired by Nippon Mining a Japanese company, but, due to a significant installed base at the U.S. Department of Defense, was forced to **sell off Gould CSD to ENCORE.**

PowerNode: a modified Concept-32

- In 1983 two Concept-32 models formed the base for the development of the PowerNode series 6000 and 9000 (ECL based).
- The Concept-32 could run a UNIX V7 port, but lacked support for VM. The PowerNodes got the hardware to support paging.

More about the hardware

- The CPU of a Concept 32/67 or a PowerNode 6000 is implemented on three boards. The boards are 38 x 46 cm(!)
- The optional Floating Point Accelerator takes another two boards.
- The CPU of the Concept 32/97 and PowerNode 9000 are on nine boards due to low density ECL technology.
- Addresses are 24 bit: max 16MB.



Channels

- Controllers on the SelBUS are **channels**, a concept found in mainframes.
- Channels process a **list of commands** that are similar for disk, tape etc.
- The **bootstrap code** is exactly the same for a tape or a disk.
- The **I/O Processor (IOP)** serves the console panel and terminal, the clock, and interfaces on the Multi Purpose (MP) bus, like a floppy disk controller, interface for serial ports or a SCSI controller.

- Channels process a **Command Chain**: a list of I/O operations.
- Part of the list can be a **Data Chain**: telling where to retrieve or deliver data.
- Data for a single I/O command can be at **scattered memory locations**.
- Example: ethernet channel can read a packet from the network and store the data in **a set of linked mbufs**.

Internal Processing Unit

- The optional second CPU is called Internal Processing Unit, **IPU**.
- A **panel switch** is used to select which of the two identical processors is to be used as CPU or IPU.
- MPX/32 and UTX/32 use the CPU & IPU config **as master / slave**, leaving the main CPU in charge of I/O.
- All **memory is shared** between CPU and IPU.
- **Two instructions**, set and clear bit in memory, include a memory lock to implement the set and release of a semaphore.
- All **I/O instructions are trapped** at the IPU. O.S. can handle the I/O at the CPU and when done execution may continue at the IPU.
- A **Signal-IPU** trap enables the operating system to signal events between CPU and IPU & vice versa.

Gould UNIX development & support

- 1984: Gould CSD took over Compion in Urbana-Champaign IL, a spinoff company of the University of Illinois' Center for Advanced Computation. This company became Gould’s UNIX R&D center. (Glenn & ~150)
- 1987: European Unix Support Center in Maarssen. (Wytze, Ted)

- Porting of BSD, dual processor implementation.
- System V release 2 compatible library and IPC.
- SUN extensions.
- Realtime extensions.
- Other projects:
  - UTX/32S C2 Secure UNIX
  - UTX/32 v3 for NP1

Features in UTX/32 by SUN Microsystems

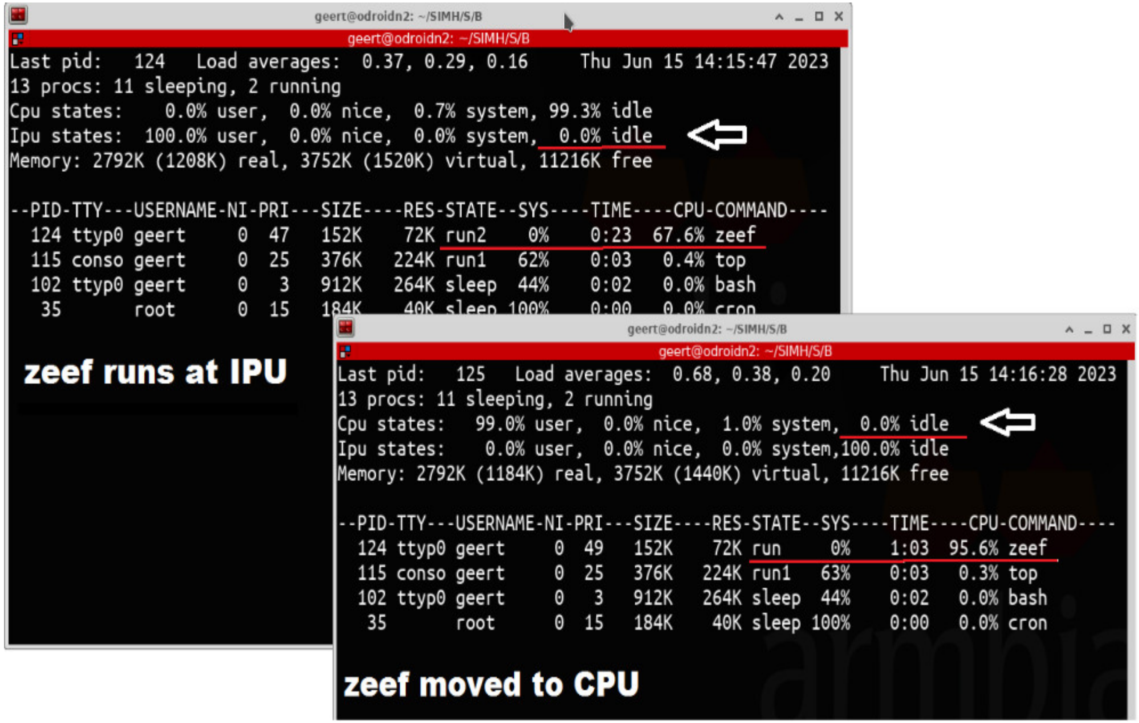
- SUN Microsystems Inc, founded by Bill Joy, contributed a lot to the development of Unix:
- Network Disk (**ND**) to provide raw disk access over the network for root and swap partitions used by SUN-2 and SUN-3 diskless workstations.
- Network File System (**NFS**).
- Yellow Pages (**YP**), later called Network Information System (NIS), to share a.o. password, group, host files in a cluster.
- External Data Representation (**XDR**), a software layer to exchange data over a network between machines with different architecture (a.o. byte order in a machine word).
- Remote Procedure Calls (**RPC**), a layer of software to support calling functions remotely over the network.

Features from AT&T System V release 2

- commands, libraries, system calls in System V style are supported in UTX/32.
- Processes in SV environment are marked and system calls are handled conform SV standards.
- Setting of $PATH is the key of chosing between environments: /usr/5bin and /usr/5lib for System V libraries.
- Using different “cc” to link with correct libraries and set type of executable.
- Use of cpio where BSD die-hards prefer tar. Assume it works.
- Basic System V release 2 style IPC features:
  - Messages
  - Semaphores
  - Shared Memory.
- *Commercial products require System V support*.

UTX/32 Real-Time Environment

- Cyclic scheduling. Cycles divided in Frames.
- High speed context switching: suspend and resume.
- Control over paging and swapping. Lock in memory after all is paged in.
- Contiguous files, deterministic transfer times.
- Support for the High Speed Data channel.
- Direct I/O including asynchronous I/O operations.
- Connected interrupts.
- Access to privileged instructions.
- Process targetting at CPU or IPU.



```
Last pid:   124   Load averages:  0.37, 0.29, 0.16     Thu Jun 15 14:15:47 2023
13 procs: 11 sleeping, 2 running
Cpu states:    0.0% user,  0.0% nice,  0.7% system, 99.3% idle
Ipu states:  100.0% user,  0.0% nice,  0.0% system,  0.0% idle
Memory: 2792K (1208K) real, 3752K (1520K) virtual, 11216K free

--PID-TTY---USERNAME-NI-PRI---SIZE----RES-STATE--SYS----TIME----CPU-COMMAND----
  124 ttyp0 geert     0  47   152K    72K run2    0%    0:23  67.6% zeef
  115 conso geert     0  25   376K   224K run1   62%    0:03   0.4% top
  102 ttyp0 geert     0   3   912K   264K sleep  44%    0:02   0.0% bash
   35       root      0  15   184K    40K sleep 100%    0:00   0.0% cron
```

zeef runs at IPU

```
Last pid:   125   Load averages:  0.68, 0.38, 0.20     Thu Jun 15 14:16:28 2023
13 procs: 11 sleeping, 2 running
Cpu states:   99.0% user,  0.0% nice,  1.0% system,  0.0% idle
Ipu states:    0.0% user,  0.0% nice,  0.0% system,100.0% idle
Memory: 2792K (1184K) real, 3752K (1440K) virtual, 11216K free

--PID-TTY---USERNAME-NI-PRI---SIZE----RES-STATE--SYS----TIME----CPU-COMMAND----
  124 ttyp0 geert     0  49   152K    72K run     0%    1:03  95.6% zeef
  115 conso geert     0  25   376K   224K run1   63%    0:03   0.3% top
  102 ttyp0 geert     0   3   912K   264K sleep  44%    0:02   0.0% bash
   35       root      0  15   184K    40K sleep 100%    0:00   0.0% cron
```

zeef moved to CPU

“Two programs, many UNIX systems” (1983)

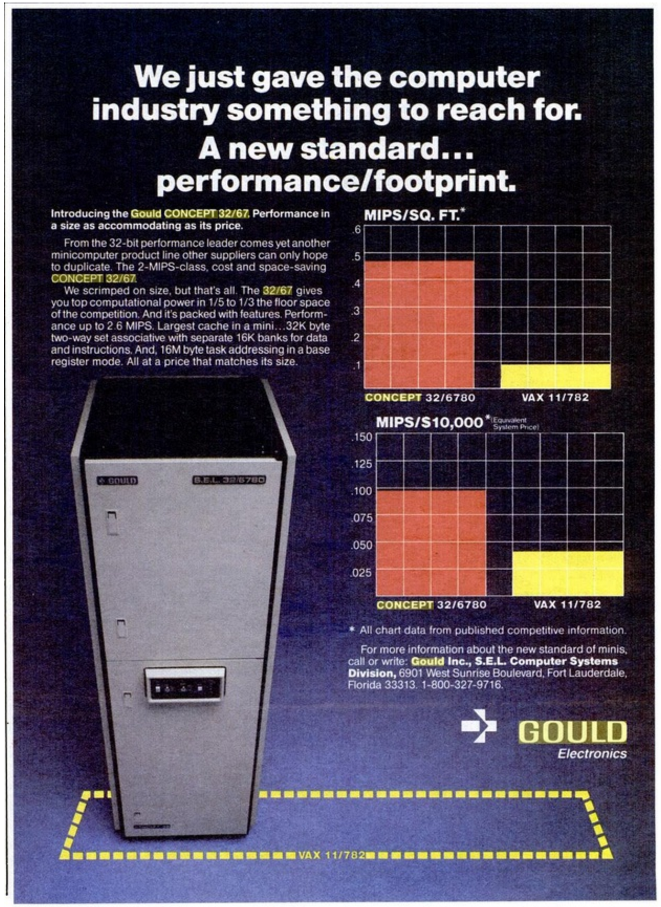
- Benchmark by Andrew Tanenbaum and Teus Hagen.

```
main()
{ TYPE i, j, k;
    for(i = 0; i < 1000; i++)
    for(j = 0; j < 1000; j++)
        k = i + j + 1983;
}
```

- EUUG Volume 3 nr 1, Spring 1983 See page 57 AUUGN-V05.4.pdf @tuhs.org

	int	short
Concept 32/87	0.9	1.5
* MicroVAX 3800	1.6	2.0
* PowerNode 6040	4.1	4.0
VAX 11/780	6.7	8.8
Concept 32/27	10.0	12.0
VAX 11/750	10.9	16.2
MicroVAX	22.2	28.2
VAX 11/730	23.9	37.4

Competing with |d|i|g|i|t|a|l| VAX 11/780



We just gave the computer industry something to reach for.

A new standard… performance/footprint.

Introducing the Gould CONCEPT 32/67. Performance in a size as accommodating as its price.

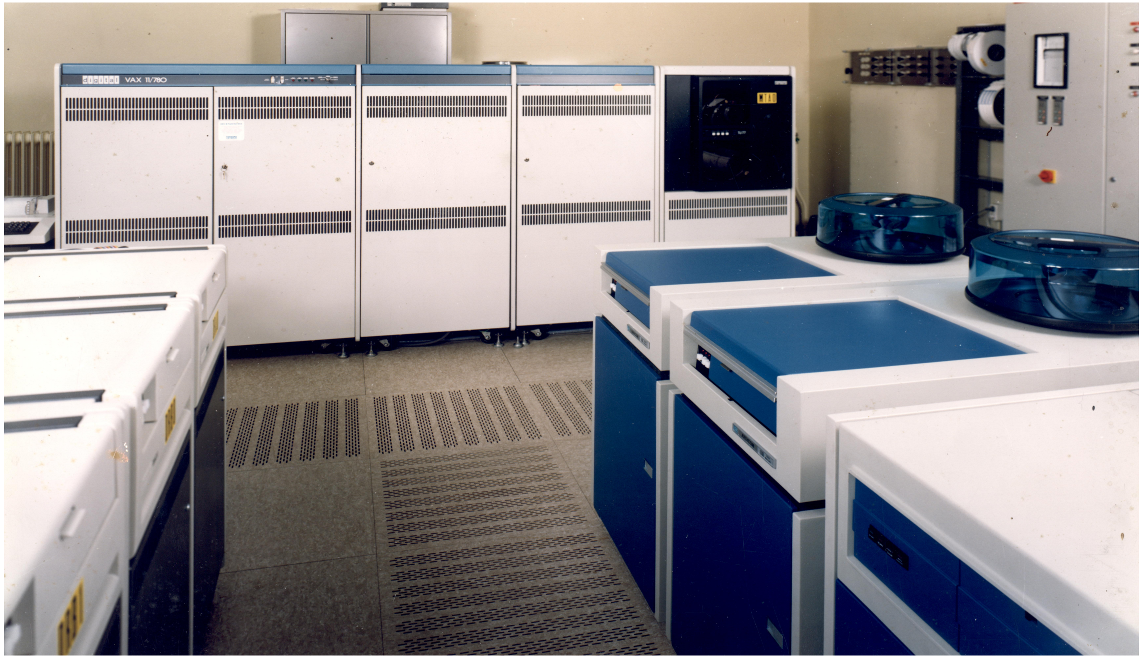
From the 32-bit performance leader comes yet another minicomputer product line other suppliers can only hope to duplicate. The 2-MIPS-class, cost and space-saving CONCEPT 32/67.

We scrimped on size, but that's all. The 32/67 gives you top computational power in 1/5 to 1/3 the floor space of the competition. And it's packed with features. Performance up to 2.6 MIPS. Largest cache in a mini…32K byte two-way set associative with separate 16K banks for data and instructions. And, 16M byte task addressing in a base register mode. All at a price that matches its size.

* All chart data from published competitive information.

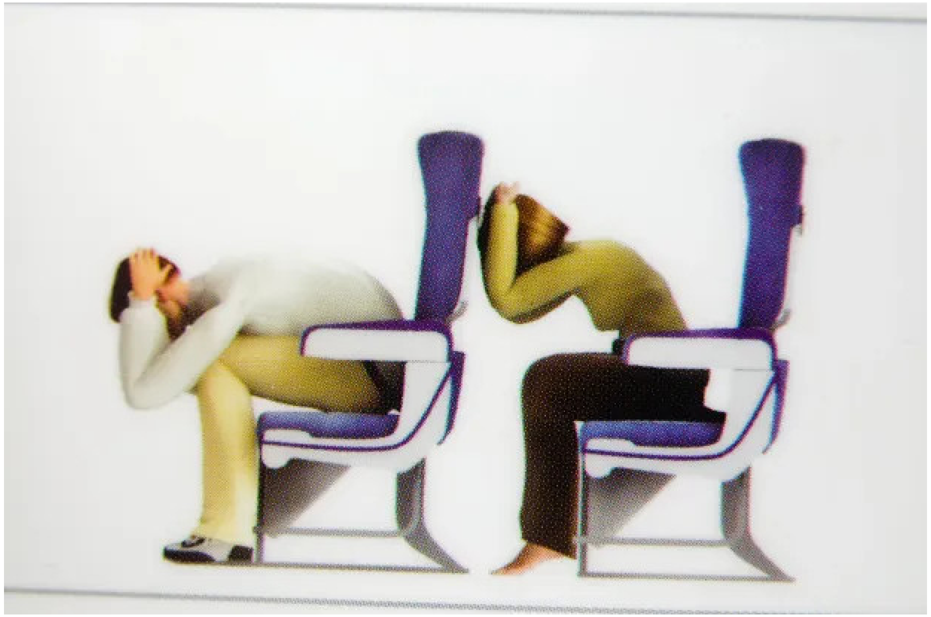
For more information about the new standard of minis, call or write: **Gould Inc., S.E.L. Computer Systems Division,** 6901 West Sunrise Boulevard, Fort Lauderdale, Florida 33313. 1-800-327-9716.

GOULD
Electronics



APT audience participation time (1)

- On what hardware did you use UNIX ***before ~1990***?
  - DEC PDP11?
  - DEC VAX?
  - Motorola 680x0?
  - SUN SPARC?
  - MIPS?
  - HP PA?
  - Intel?



APT audience participation time (2)

- What do you know about the hardware you run [UL]i*n[iu]x on today?
  - sizeof(long)? 4? 8?
  - How many registers?
  - Big-endian or little-endian?
  - Know (some) assembler mnemonics?
- *[UL]i*n[iu]x guys ‘n girls don't know much about the hardware they use.* ***T or F?***

Where were Concepts & PowerNodes used in NL?

- NIKHEF: 32/87, 9080, NP1
- Bureaus voor Rechtshulp: 12x 6040
- Hogeschool Den Haag: 9080
- Blindenbibliotheken: 2x 6040
- KLM Schiphol, Eelde: Concepts
- Philips Semiconductor Nijmegen: Concept
- Fokker: Concepts
- ECN Petten
- TNO Delft
- ESA/Estec
- TU Delft
- NLR
- PGEM

Concept 32: ~70% market share in Flight Simulators

Mid 90s: end of life.

- Developments in UNIX that did not make it to UTX/32:
  - **Shared libraries and dynamically linked objects** in System V release 4 (Usenix, 1986, Atlanta) and SunOS 4.x (Usenix, 1987, Phoenix).
  - **Virtual I/O**. Paging from/to a file. Copy a large file to /dev/null in a split second.
- **ANSI C & newer**. No gcc, although there has been GDB for UTX/32.
- **16MB max memory** became a show stopper.
- **Floating Point** in IBM/360 style; incompatible with IEEE 754,
- Early 90s : fast (RISC) processors get to the market: DEC Alpha AXP, IBM POWER, SUN SPARC, MIPS etc.

SIMH: the simulation of historical computers

- SIMH (CHM Boston, Bob Supnik) is about simulation of old computers.
- Reliving the old operating systems of long gone computers.
- Preserving old manuals and software.
- SIMH is serialising all that happens in parallel in real hardware.
- You will find simulators for a large range of old DEC, IBM, HP, DG etc in the SIMH archives at GitHub.
- You need detailed knowledge to build a SIMH simulator for a computer.
- If there is one person, who has climbed deep down into the SEL-32 systems, it is Jim Bevier.
- Jim has worked for Systems Engineering Laboratories from 1976 until 1983. Later, working for another company, he has done more MPX/32 system software.
- In 2012 Jim wanted to run a few saved programs on a Linux machine at home and started developing an emulator and several cross tools for Linux. After his retirement in 2013 work continued.

Milestones in SIMH/sel32 development (1)

- December 19, 2018: First release of Jim’s SIMH/sel32 simulator. It ran MPX/32 1.5F.
- May 19, 2019: my first contact with Jim. The tape (½” open reel) with diagnostics converted to SIMH .tap format.
- Jim worked through the set of CPU diagnostics verifying and debugging his simulator.
- Jan 14, 2020: Jim reports UTX/32 2.1A distribution tape booting up to the # prompt using a RAM disk only.
- Disks not yet working.

Disks are not that easy!

- To replace bad blocks, disks have a few spare blocks ***per track*** as opposed to a spare track at the end of a disk.
- So, we need book-keeping per track: track labels in addition to sector labels.
- Disk preparation (including format, verify, tracklabels, partitions) is a major step in the installation of UTX.
- *Without all this magic UTX won’t install.*

Milestones in SIMH/sel32 development (2)

- Jan 25, 2020: the benchmark “zeef” compiled at the real PowerNode and put on a dump tape, loaded into the memory disk on the simulator after booting up from the boottape.
- July 2020: Jim managed to recreate an MPX/32 3.4 tape from bits and pieces and boot and run it.
- /usr/games/rain used as stress test for console output.
- September 2020 the network is up and running.
- November 2020: the file UTX32-v21B-Binary-Kit.tgz uploaded to bitsavers.org. This is starting to make sense.
- June 2021: install and run MPX/32 3.6 from a hand built Software Distribution Tape.
- March 2022: SIMH/sel32 released on GitHub to run MPX/32 and UTX/32.

The “ZEEF” benchmark

- “Zeef”: finding primes using the Sieve of Erathostenes.
  - Static table with 10,000 prime numbers.
  - Calculate the next 4,000 primes.
  - It takes 36,384,896 calculations.
- *Awaiting Raspberry Pi 5.*

	secs	rating
PowerNode 6040	1468	1
Raspberry Pi 3B+	461	3.18
IBM xSeries 345	213	6.89
Raspberry Pi 4B	186	7.89
Odroid N2+	133	11.04
AMD FX-6300	50	29.36

Experimental implementation of the IPU

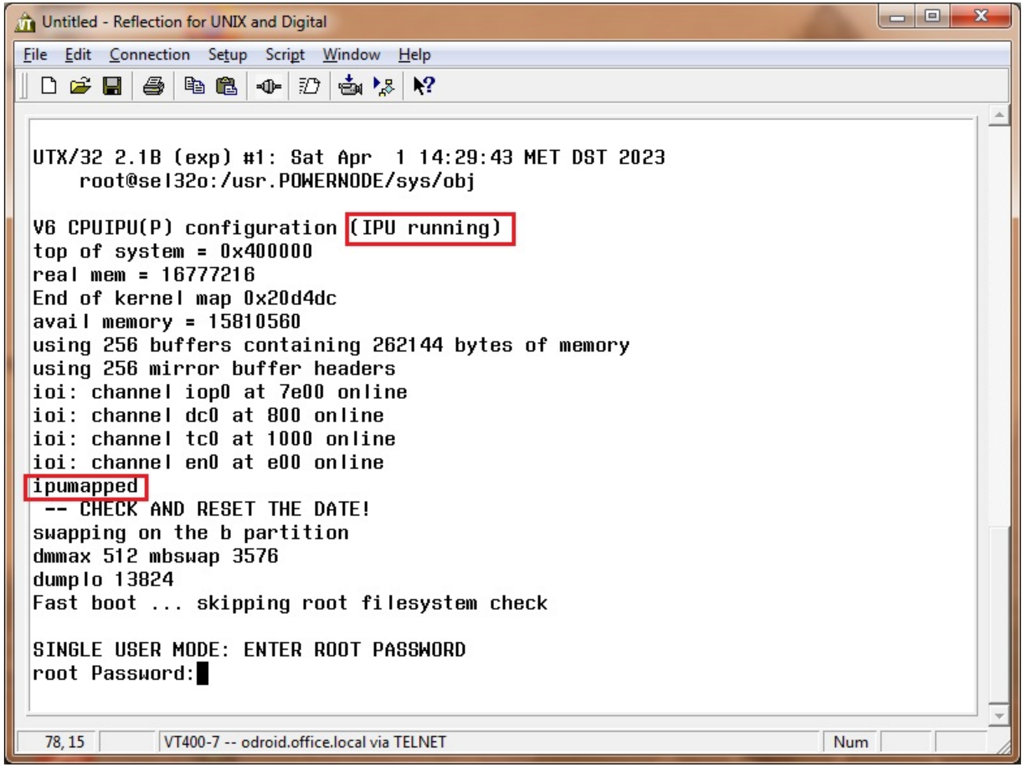
- Idea: fork() a second SIMH/sel32 just before it attached peripherals. Use SIMH/sel32 as a blackbox to run the IPU.
- All memory created by mmap; no longer a static array.
- Using POSIX semaphores to make the set and clear bit instructions atomic between CPU and IPU.
- Kernel of UTX/32 assumed ready for use of IPU.
- MPX/32 wasn’t, but Jim was the lead programmer himself.
- At the IPU, traps and system-calls cause a hand-off to CPU. Very little kernel code for IPU.
- CPU loads and boots the kernel.
- Boot-time puzzles: how to start the IPU and have it wait for the CPU.

CPU and IPU behave differently,

- HALT still listens for Signal-IPU traps, a detail that I was not aware of and was not implemented in the single-cpu version.
- Ten I/O instructions had to be changed to give an “IPU undefined instruction” trap.
- CPU/IPU trap and interrupt vectors are at different memory locations.
- SIMH should run on Windows platform too: no fork(2)… using pthreads.
- The two instructions to set and release a semaphore bit use Mutex in the threaded version.
- *Modern multi-core platforms give you a double speed.*

Q1/2023: the implementation of the IPU is finished.

- The threaded IPU is about 7 % faster than the CPU. Checking for SIMH events such as interrupts from channels done in the CPU-thread only.
- In the forked version CPU and IPU perform equal. Both check for events which only happen at the CPU.
- ***The return of the fire breathing dragons has completed.***


Untitled - Reflection for UNIX and Digital
File
Edit
Connection
Setup
Script
Window
Help
78, 15
VT400-7 -- odroid.office.local via TELNET
Num

```
UTX/32 2.1B (exp) #1: Sat Apr  1 14:29:43 MET DST 2023
    root@sel32o:/usr.POWERNODE/sys/obj

V6 CPUIPU(P) configuration (IPU running)
top of system = 0x400000
real mem = 16777216
End of kernel map 0x20d4dc
avail memory = 15810560
using 256 buffers containing 262144 bytes of memory
using 256 mirror buffer headers
ioi: channel iop0 at 7e00 online
ioi: channel dc0 at 800 online
ioi: channel tc0 at 1000 online
ioi: channel en0 at e00 online
ipumapped
 -- CHECK AND RESET THE DATE!
swapping on the b partition
dmmax 512 mbswap 3576
dumplo 13824
Fast boot ... skipping root filesystem check

SINGLE USER MODE: ENTER ROOT PASSWORD
root Password:█
```

EOT… any questions?

https://geerol.home.xs4all.nl/sel32.htm