Soul of an Old Machine: DPS8/M Emulator Project

Bring Multics back to life

Multics Timeline

1965 -- Joint Project of MIT Project MAC, Bell Telephone Laboratories, General Electric Company Large Computer Products Division

1967 -- First light: Multics boots single user mode

1969 -- MIT starts providing timesharing service on Multics; Bell drops out and UNIX starts

Mid 1980s -- Honeywell stops Multics development; almost 100 sites

1988 -- MIT shuts down their Multics system

1992 -- MR12.5 release

2000 -- Last Multics System shutdown

Multics Hardware Timeline

1960s -- The GE-635

1967 -- 645 delivered to Project MAC “later than planned”

1970 -- Honeywell buys GE LCS

1973 -- Honeywell ships the 6180

1977 -- 6180 rebranded as Level 68, some performance improvements

1979 -- Level 68 rebranded as DPS8, some performance increase, a few architectural changes

Multics Operating System

- Segmented memory
- Virtual memory
- High-level language implementation
- Shared memory multiprocessor
- Multi-language support
- Relational database
- Hierarchical file system
- Security
- Online reconfiguration
- Hierarchical file system
- Dynamic linking
- Command language

The Problem

Multics ran only on this hardware, and there are no remaining running instances of the hardware.

Solution 1: Port Multics to new hardware

Solution 2: Write an emulator

Multics Hardware

6180

Level 68

DPS8/M

Honey

Honeywell

Multics Hardware

36 bit word, 18 bit addresses, 15 bit segment numbers

Modular design

- CPU Central Processing Unit
- SCU System Control Unit: memory, interrupt management
- IOM Input/Output Manager: 56 controller channels
- Disks, drums, tapes, printers, card readers/punches
- FNP Front end Network Processor
- ABSI IMP interface
- Operators console

Multics Terminology

Segment

Ring

The Pieces

Bitsavers collection of scanned documents

Multicians

The Pieces

Multicians.org

- preserve the technical ideas and advances of Multics so others don't need to reinvent them
- record the history of Multics, its builders, and its users before we all forget
- give credit where it's due for important innovations
- remember some good times and good people.

Overview of DPS8/M emulation history

2007 Orangesquid “Honeywell 6180 (DPS-8) Machine emulator”

2007 Bull releases Multics source

2008 Michael Mondy “multics-emul”

2012 doon386 “dps8m”

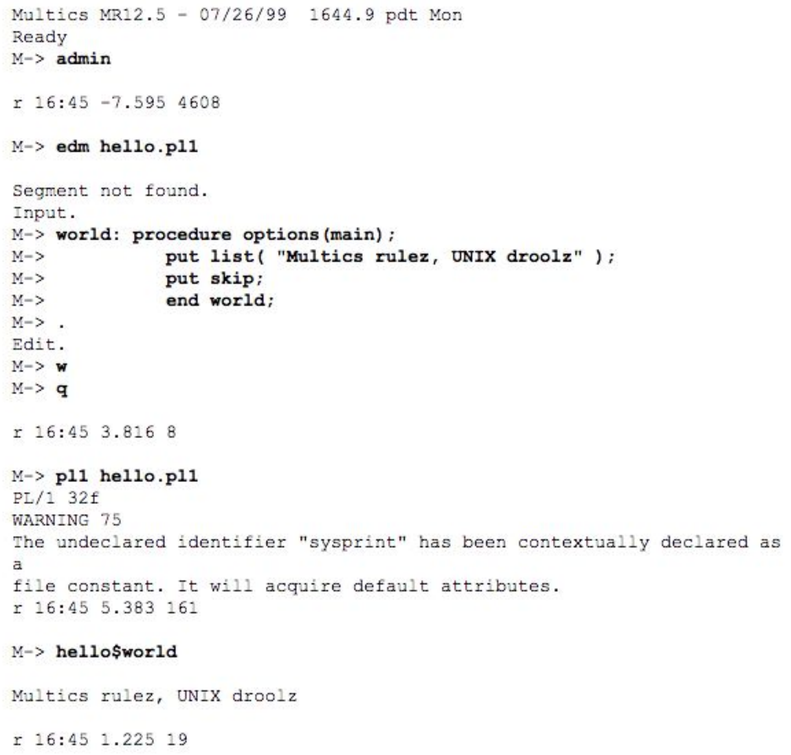
Nov 2014 MRUD, dps8m release 1 Alpha

Dec 2014 Live Demo

Dec 2016 MR12.6

Aug 2017 Version 1.0

MRUD



```
Multics MR12.5 - 07/26/99  1644.9 pdt Mon
Ready
M-> admin

r 16:45 -7.595 4608

M-> edm hello.pl1

Segment not found.
Input.
M-> world: procedure options(main);
M->              put list( "Multics rulez, UNIX droolz" );
M->              put skip;
M->              end world;
M-> .
Edit.
M-> w
M-> q

r 16:45 3.816 8

M-> pl1 hello.pl1
PL/1 32f
WARNING 75
The undeclared identifier "sysprint" has been contextually declared as
a
file constant. It will acquire default attributes.
r 16:45 5.383 161

M-> hello$world

Multics rulez, UNIX droolz

r 16:45 1.225 19
```

My history with the emulator

2013 Optimism
2014 Reality check

Progress

IOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

T&D

ISOLTS

WAM

Progress

IOM Emulation

Feb 6, 2014 “The great IOM rewrite, try 3.”
Jun 23, 2014 “The great IOM rewrite, try #4”
Jun 24, 2014 “Revert IOM rewrite try4.”
Jun 26 2014 “Rewrite on IOM, try #5”
Aug 3,2014 “IOM-B paging support”
Oct 10,2015 “Rewrite IOM”

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

Progress

IOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

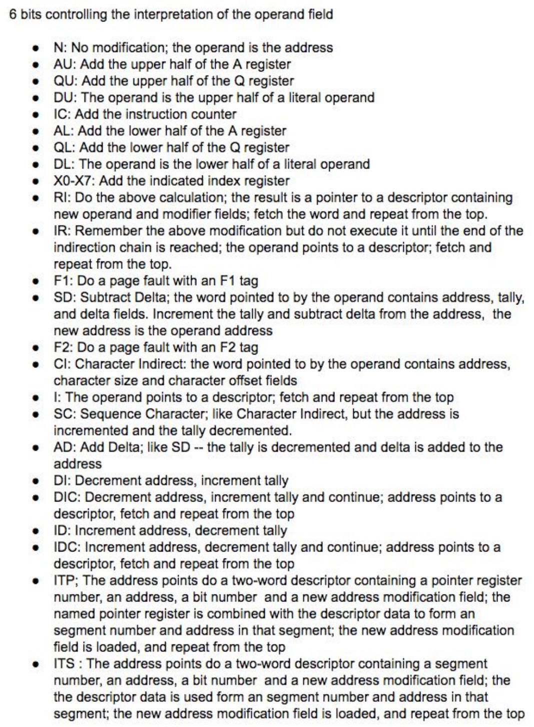
Append Unit

T&D

ISOLTS

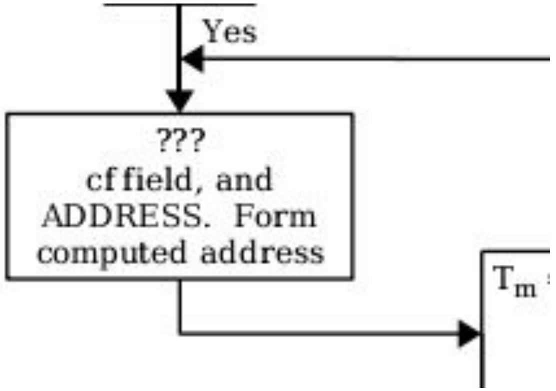
WAM

Address Modifier Field


6 bits controlling the interpretation of the operand field

- N: No modification; the operand is the address
- AU: Add the upper half of the A register
- QU: Add the upper half of the Q register
- DU: The operand is the upper half of a literal operand
- IC: Add the instruction counter
- AL: Add the lower half of the A register
- QL: Add the lower half of the Q register
- DL: The operand is the lower half of a literal operand
- X0-X7: Add the indicated index register
- RI: Do the above calculation; the result is a pointer to a descriptor containing new operand and modifier fields; fetch the word and repeat from the top.
- IR: Remember the above modification but do not execute it until the end of the indirection chain is reached; the operand points to a descriptor; fetch and repeat from the top.
- F1: Do a page fault with an F1 tag
- SD: Subtract Delta; the word pointed to by the operand contains address, tally, and delta fields. Increment the tally and subtract delta from the address, the new address is the operand address
- F2: Do a page fault with an F2 tag
- CI: Character Indirect: the word pointed to by the operand contains address, character size and character offset fields
- I: The operand points to a descriptor; fetch and repeat from the top
- SC: Sequence Character; like Character Indirect, but the address is incremented and the tally decremented.
- AD: Add Delta; like SD -- the tally is decremented and delta is added to the address
- DI: Decrement address, increment tally
- DIC: Decrement address, increment tally and continue; address points to a descriptor, fetch and repeat from the top
- ID: Increment address, decrement tally
- IDC: Increment address, decrement tally and continue; address points to a descriptor, fetch and repeat from the top
- ITP; The address points do a two-word descriptor containing a pointer register number, an address, a bit number and a new address modification field; the named pointer register is combined with the descriptor data to form an segment number and address in that segment; the new address modification field is loaded, and repeat from the top
- ITS : The address points do a two-word descriptor containing a segment number, an address, a bit number and a new address modification field; the the descriptor data is used form an segment number and address in that segment; the new address modification field is loaded, and repeat from the top

Multics



Multics

Progress

IIOM Emulation

Computed Address Formation

Addressing Modes

- Absolute
- BAR
- Append

Faults and Interrupts

Append Unit

T&D

ISOLTS

Progress

IIOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

T&D

ISOLTS

WAM

Progress

IIOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

T&D

ISOLTS

WAM

```
       |
       V
  -----------
/  APU data    \  No ----+
\  movement?   /         |
  -----------            |
       |   Yes           |
       V                 |
--------------           |
| Load/store |           |
|  APU Data  |           |
--------------           |
       |                 |
       |<----------------+
       V
```

Progress

IIOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

T&D

ISOLTS

WAM

Progress

IIOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

T&D

ISOLTS

WAM

Progress

ISOLTS

- Takes the CPU to be tested offline
- Reserves the memory in one of SCUs
- Reconfigures the test CPU to use that SCU as its low memory
- Installs Test and Diagnostic code in that memory
- Starts the test CPU running the tests

Progress

IIOM Emulation

Computed Address Formation

Addressing Modes

Faults and Interrupts

Append Unit

T&D

ISOLTS

WAM

Other documentation highlights

PR/AR Dichotomy

Other documentation highlights

ABSA

If the absa instruction is executed in absolute mode, C(A) will be undefined.

Attempted execution in normal or BAR modes causes an illegal procedure fault

Other documentation highlights

IOM Channel Status Data Format

If M is 0, see I

If I is 0, see M

Other documentation highlights

Bitstring Operand Descriptor

The IC modifier is permitted in MFk.REG and

C (OD)32,35 only if MFk.RL = 0

Other documentation highlights

Bitstring Operand Descriptor

(MFk.REG and C (od)32,35) only if MFk.RL = 0

vs:

MFk.REG and (C (od)32,35) only if MFk.RL = 0)

Other documentation highlights

Sept 21, 2017

“Blubber, blubber, blubber, cry, whimper.....”

Other documentation highlights

Sept 21, 2017

“Blubber, blubber, blubber, cry, whimper.....”

"EMULATING A HONEYWELL 6180 COMPUTER SYSTEM"

Other documentation highlights

```
prime results    ar0

          s/b 77777663
          was 77777706
secondary results   x
              s/b 000000
              was 000000
function in error -
test execution of eis instruction s6bd.
prime results of ar0 will be in error if change
order phaopa087 is not installed.
```

Adventures in Third Party Libraries

Leap seconds

Wins

MCRB

Exhausting workday

Multician status

The maintenance panel

Systems

Tipping Points

Wins

```
2 set unaligned,  /* SPECIFIES WHICH ATTRIBUTES TO SET */
      3 (allow_write,
         tms,
         tus,
         tpd,
         audit,
         explicit_deactivate_ok) bit (1),
      3 pad bit (39),
```

“I like the idea of submitting a bug report 14 years after the last site was shutdown. Might be a record.

Do we have to reconvene the MCR board to approve the fix?”

Wins

MCRB

Exhausting workday

Multician status

The maintenance panel

Systems

Tipping Points

“Super. Nothing like coming home from an exhausting workday to find a bug in my 35-year-old code has come back to haunt me. “

My proposed fix was accepted as MCR10010 and shipped in MR12.6e.

Wins

MCRB

Exhausting workday

Multician status

The maintenance panel

Systems

Tipping Points

Wins

MCRB

Exhausting workday

Multician status

The maintenance panel

Systems

Tipping Points

Wins

MCRB

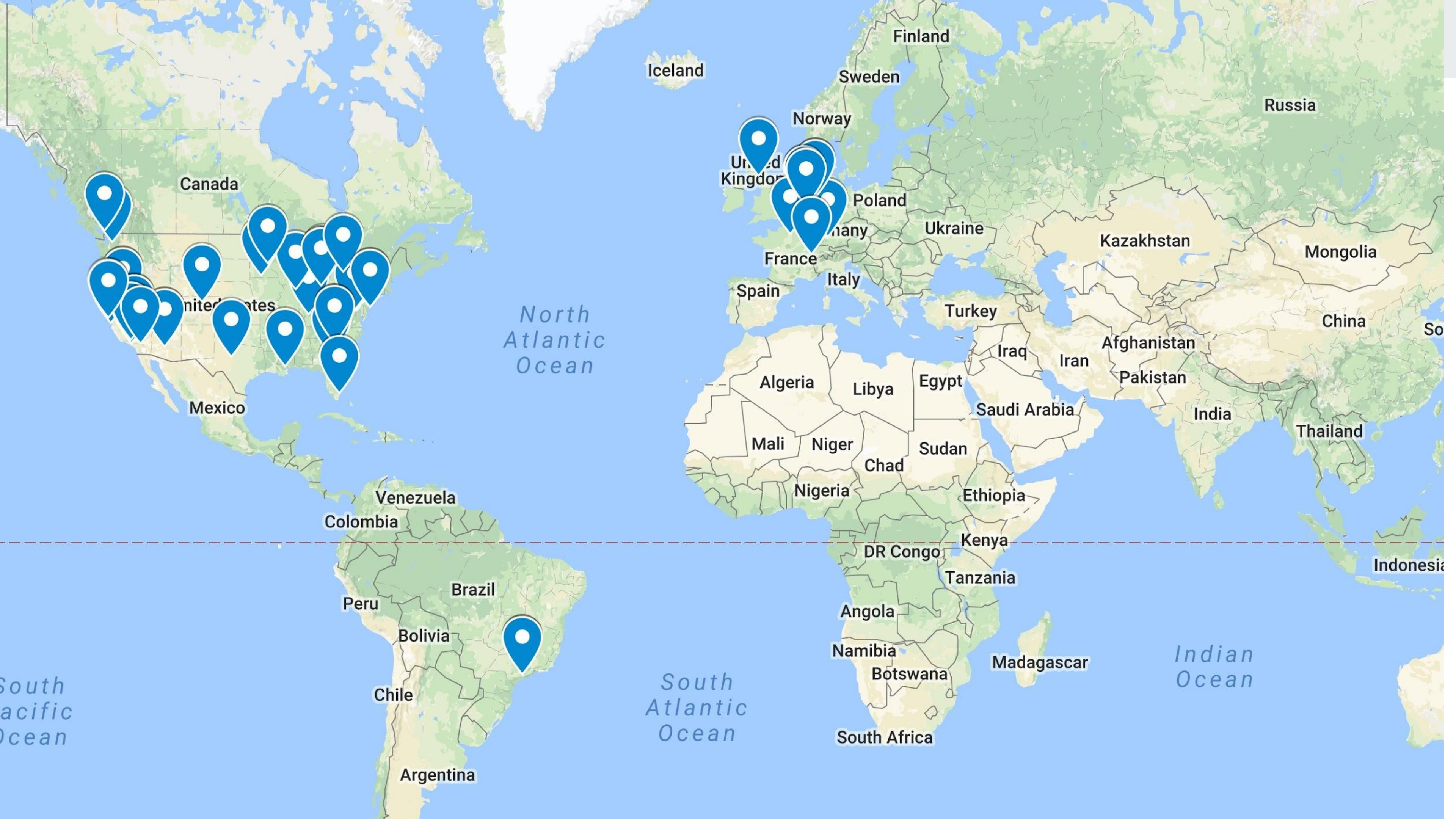
Exhausting workday

Multician status

The maintenance panel

Systems

Tipping Points


Finland
Iceland
Sweden
Russia
Norway
United Kingdom
Canada
Poland
Ukraine
Germany
Kazakhstan
Mongolia
France
Italy
Spain
United States
North Atlantic Ocean
Turkey
China
Afghanistan
Iraq
Iran
Pakistan
Algeria
Libya
Egypt
Mexico
Saudi Arabia
India
Thailand
Mali
Niger
Sudan
Chad
Nigeria
Ethiopia
Venezuela
Colombia
Kenya
DR Congo
Indonesia
Tanzania
Brazil
Peru
Angola
Bolivia
Namibia
Madagascar
Indian Ocean
Botswana
South Atlantic Ocean
South Pacific Ocean
Chile
South Africa
Argentina

Wins

MCRB

Exhausting workday

Multician status

The maintenance panel

Systems

Tipping Points

Current Status

Release 1.0, bugfix release 1.0.1 imminent

87K lines of C

Decimal math library (DECPUN): 20K

Event library (libuv): 2K

Total: 105K

Multics is at release 12.6f

The Future

Threaded support for Multiple CPUs

FNP emulation

Fixing Append Cycle/Computed Address Formation

Pl/I compiler

Networking

Code improvement

Acknowledgements

My fellow DPS8/M developers past and present

The Multicians

Bull HN Information Systems

SIMH

SourceForge and Wikidot

Classic Computers mailing list members

Living Computer Museum + Labs

Sultan Library, Gathering Grounds Coffeehouse and Galaxy Chocolates

Damon and Lily

Neil, Justin and Gerald

Soul of an Old Machine: DPS8/M Emulator Project

Bring Multics back to life