History of the mainframe

From S/360 to Linux

Claudio Imbrenda, Nico Böhr

IBM

21. Gulaschprogrammiernacht
June 9, 2023

Whoarewe

Claudio Imbrenda
imbrenda@de.ibm.com

KVM s390 Co-maintainer
KVM-unit-tests s390 Co-maintainer

Nico Böhr
Nico.Boehr@ibm.com

KVM s390 Developer
KVM-unit-tests s390 Co-maintainer

Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:
IBM, the IBM logo, IBM Z, IBM z Systems, IBM z14, WebSphere, DB2 and Tivoli are trademarks of IBM Corporation in the United States and/or other countries. For a list of additional IBM trademarks, please see https://ibm.com/legal/copytrade.shtml.

The following are trademarks or registered trademarks of other companies: Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both Microsoft, Windows,Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries or both. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc. InfiniBand is a trademark of the InfiniBand Trade Association. Other company, product, or service names may be trademarks or service marks of others.

Linux penguin image courtesy of Larry Ewing (lewing@isc.tamu.edu) and The GIMP

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment. IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information is provided “AS IS” without warranty of any kind. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area. All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Overview

- Historic background and birth of the mainframe
- S/360 hardware features
- S/360 software
- S/370 hardware and software – virtual memory and virtualization
- S/390 – CMOS mainframes
- z/Architecture – modern 64-bit
- Linux on the mainframe

State of the art in the 50s

- Many different incompatible systems, even from the same vendor
- Each system had only few small variations
  - Amount of memory, speed, number of I/O peripherals
- Each system had its own incompatible operating system
  - Closely modelled after the hardware
  - Drivers often rewritten from scratch
- Moving software between different systems required rewriting it
  - Different OS interfaces, programming languages, machine code
- Vendors had to support several different platforms

Bet the company

- Strong competition, need for innovation
    - IBM risked to become “a company that sells computers” like many others
- A task group was created to address the issue
    - It recommended developing five compatible systems spanning a 200-fold performance range
- IBM followed the advice and replaced the whole product line with **compatible machines**
- Estimated total cost 675 millions, of which 30 millions for software

Bet the company

- Strong competition, need for innovation
    - IBM risked to become “a company that sells computers” like many others
- A task group was created to address the issue
    - It recommended developing five compatible systems spanning a 200-fold performance range
- IBM followed the advice and replaced the whole product line with **compatible machines**
- Estimated total cost 675 millions, of which 30 millions for software
- Ended up costing 5 billions! (500 millions of which for software)
- Toward the end of the project, IBM was in financial difficulties

Innovations

- 8-bit bytes
- Instruction Set Architecture (ISA)
- Microcode for commercial computers
- Solid Logic Technology (SLT)
- Hardware abstraction in the OS

S/360 architecture

- Big endian
- 24-bit addresses
- Consistent instruction formats
  - 2, 4, or 6 bytes long; first 2 bits of instruction indicate length
- Registers:

16	32-bit	General purpose	
1	64-bit	Program Status Word	
4	64-bit	Floating point	(optional)

- Channel I/O
- Interrupts (with classes and subclasses)
- Protection (storage keys, optional)
- Floating point (not IEEE, optional)
- Decimal (BCD) arithmetic (optional)
- Dynamic Address Translation (virtual memory, S/360-67)
- Multiprocessing (S/360-65)

Protection (Storage keys)

Storage keys are 4-bit values, optionally with a 5th bit for fetch protection.

Two instructions allow to set and get the storage keys for each *2kB* block of real (physical) memory.

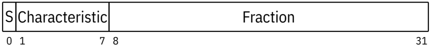
The PSW also has a key. At each memory access the key of the accessed memory block is compared with the key in the PSW.

	Storage key x		Storage key y no fetch protection		Storage key y fetch protection*	
	read	write	read	write	read	write
PSW key 0	√	√	√	√	√	√
PSW key x	√	√	√	X	X	X

* only with Fetch-protection feature

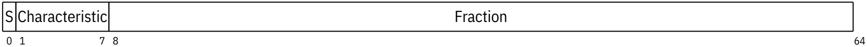
Hexadecimal floating point

Short Floating-point


S | Characteristic | Fraction

0 1 7 8 31

Long Floating-point


S | Characteristic | Fraction

0 1 7 8 64

- Not IEEE compatible
  - The IEEE 754 standard was released only in 1985, 20 years after S/360!
- Long and short only differ in size of fraction
- Characteristic: Biased exponent (0..127 = -64..+63)
- Base 16: $value = 16^{exponent} \cdot 0.fraction$
- No NaNs, no infinities
- Some special conditions can raise (maskable) program interrupts

S/360

Model	Shipped	kIPS	Memory (kiB)	Weight (kg)
30	1965	10	8–64	771
		⋮		
195	1971	10000	1024–4096	6101–12859

Performance calculated (not measured) based on a mix of instructions typical of scientific (*Gibson Mix*).

S/360-67

- New PSW Format
- Control registers
- Virtual memory (24 and 32 bit)
  - 4kB pages
  - TLB

S/360-67

- New PSW Format
- Control registers
- Virtual memory (24 and 32 bit)
  - 4kB pages
  - TLB (8 entries!)

S/360 Operating systems

1965	BOS/360	Basic Operating System
1965	TOS/360	Tape Operating System
1966	DOS/360	Disk Operating System
1966	OS/360	Operating System (PCP and MFT)
1966	RAX	Remote Access Computing
1967	CMS	Cambridge Monitor System
1967	CP-40, CP-67	Control Program
1967	TSS/360	Time Sharing System
1967	OS/360	Operating System (MVT)
1967	ORVYL	
1968	CALL/360	
1968	ACP	Airline Control Program
1968	VP/CSS	

BPS/BOS/TOS/DOS

- The development of OS/360 was lagging behind due to complexity
- Interim smaller OSs developed to fill the gaps
- OS/360 would not run on smaller systems, contrary to stated goals
- The smaller OSs were needed for smaller machines
- Customers invested in DOS and did not want to switch to OS/360

OS	Memory	Tape	Disk	Year	Notes
BPS	8kB	(opt)	–	1964	not an actual OS
BOS/360	8kB	–	yes	1965	
TOS/360	16kB	yes	–	1965	
DOS/360	16kB	–	yes	1966	

BPS	IBM Basic Programming Support/360
BOS	Basic Operating System/360
TOS	Tape Operating System/360
DOS	Disk Operating System/360

OS/360

- The flagship OS for S/360 mainframes
- Three variants, sharing API, ABI, and Job Control Language
  - PCP – single task (48kB)
  - MFT – fixed number of tasks (256kB)
  - MVT – variable number of tasks (512kB)
- Memory partitions for user programs
- File name structure, allowing for hierarchies
- Various forms of remote access
- Sub-tasks (not PCP) – threads within one job
- Toleration for S/370

CP/CMS

Cambridge Monitor System

- Single user
- Can run on a bare-metal S/360

CP/CMS

Cambridge Monitor System

- Single user
- Can run on a bare-metal S/360

Control Program (CP-40)

- Virtual machine (Trap and emulate)
- Ran only on one specially modified S/360-40
- Research prototype for the upcoming S/360-67
- Multiuser support for CMS!

CP/CMS

Cambridge Monitor System

- Single user
- Can run on a bare-metal S/360

Control Program (CP-40)

- Virtual machine (Trap and emulate)
- Ran only on one specially modified S/360-40
- Research prototype for the upcoming S/360-67
- Multiuser support for CMS!

Control Program (CP-67)

- S/360-67 port of CP-40
- Support more VMs
- Used in production
- Not officially supported by IBM
- Later versions support virtual memory in the guest

CP/CMS

Cambridge Monitor System

- Single user
- Can run on a bare-metal S/360

Control Program (CP-40)

- Virtual machine (Trap and emulate)
- Ran only on one specially modified S/360-40
- Research prototype for the upcoming S/360-67
- Multiuser support for CMS!

Control Program (CP-67)

- S/360-67 port of CP-40
- Support more VMs
- Used in production
- Not officially supported by IBM
- Later versions support virtual memory in the guest

VM/370

- Re-implementation of CP for S/370

ACP – Airline Control Program

- PARS – Programmed Airline Reservations System (1968)
  - Consolidate existing airline reservation systems
  - SABRE, Deltamatic, PANAMAC
- ACP split from PARS (1969)
- Real-time
- Transaction-oriented
- Not a general purpose OS
- Later also used by banks
- TPF — Transaction Processing Facility
  - z/TPF — 64-bit extension

S/370 architecture

Base S/370 architecture (1970)

- **Translation (virtual memory)**
  - Different and incompatible with S/360-67!
  - 2kB and 4kB page sizes
- Extended-precision floating point
- Dual address space
- Support for multiprocessing

S/370 architecture

Base S/370 architecture (1970)

- **Translation (virtual memory)**
  - Different and incompatible with S/360-67!
  - 2kB and 4kB page sizes
- Extended-precision floating point
- Dual address space
- Support for multiprocessing

S/370-XA (Extended Architecture, 1983)

- Switchable per-process 31-bit mode (new PSW bit)
- Only 4kB supported for storage keys and virtual memory pages
- Channel I/O completely rehauled
- Vector instructions (3090 only)

S/370 architecture

Base S/370 architecture (1970)

- **Translation (virtual memory)**
  - Different and incompatible with S/360-67!
  - 2kB and 4kB page sizes
- Extended-precision floating point
- Dual address space
- Support for multiprocessing

S/370-XA (Extended Architecture, 1983)

- Switchable per-process 31-bit mode (new PSW bit)
- Only 4kB supported for storage keys and virtual memory pages
- Channel I/O completely rehauled
- Vector instructions (3090 only)

ESA/370 architecture (Enterprise Systems Architecture, 1988)

- Access register mode
- Home address space
- LPAR – Logical partitions

Virtualization

IBM starts to recognize the importance of virtualization.

Virtual-Machine Assist (1980)

- For S/370 without XA
- A collection of 6 assists for VM/370

START INTERPRETIVE EXECUTION (1984)

- For S/370-XA
- Nested paging supported since the beginning
- Almost all instructions execute without exit
- A control block in memory describes a guest CPU
- Still there today!

Clones

Most S/360 clones were not drop-in replacements

- Mostly copying the instruction set, or subsets
- Not aiming to perfect compatibility
  - Except for the Soviet ES EVM

S/370 clones were meant as drop-in replacements

- Amdahl Corporation
  - Gene Amdahl, former IBMer
  - Started selling drop-in replacements for IBM mainframes
- Many companies followed suit
  - Fujitsu, Hitachi, ES EVM, Magnuson Computer Systems, Mitsubishi, Siemens, Univac
- Some competitors at times sold better hardware than IBM!

Operating systems overview

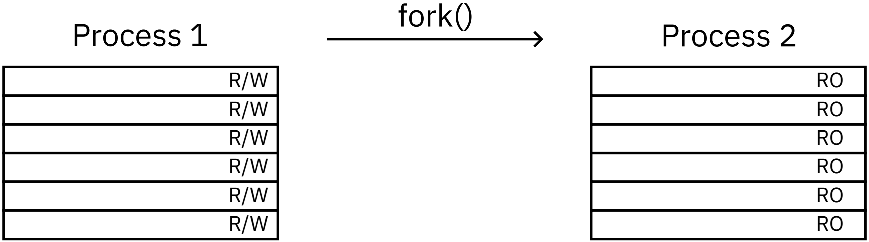
1965	DOS/360			
1966		OS/360 PCP, MFT		
1967		OS/360 MVT	CP/CMS	ACP
1968		OS/360 MFT II	VP/CSS	
1972	DOS/VS	OS/VS1	VM/370	
1972		OS/VS2R1 (SVS)		
1974		OS/VS2R2 (MVS)		
1978		MVS/SE		
1979	DOS/VSE			TPF
1980		MVS/SP	VM/SP	
1983		MVS/XA		
1984			VM/XA	
1986	VSE/SP			
1988		MVS/ESA		
1990	VSE/ESA		VM/ESA	
1995		OS/390		
2000		z/OS	z/VM	
2005	z/VSE			z/TPF
2021	VSE^n			

1980	Unix	[tss]
1980	UTS$^{(amdahl)}$	[vm]
1984	IX/370	[tss,vm]
1985	VM/IX	[vm]
1988	AIX/370	[vm]
1991	AIX/ESA	
1993	MVS/ESA OpenEdition	[os]
1999	Linux	
2001	z/OS UNIX System Services	[os]
2008	OpenSolaris	[vm]

S/390 architecture

- Additional floating point registers (16 in total)
- Binary floating point (IEEE 754)
- Instructions to load and work with immediate values and relative addresses
- Suppression on protection

AIX/ESA - Suppression on Protection



Process 1

R/W
R/W
R/W
R/W
R/W
R/W

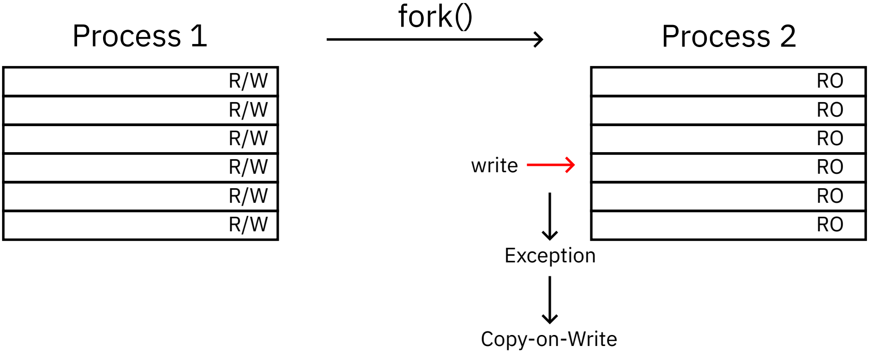
fork()

Process 2

RO
RO
RO
RO
RO
RO

- `fork()` creates identical copy of process
- memory is not copied → all pages read-only

AIX/ESA - Suppression on Protection



- write to read-only page of process 2 causes exception → page copied
- Instruction causing exception must not have had any side-effects
- s390: write crossing page boundary may execute partly!
- Feb 1993: *Suppression-on-protection* for AIX/ESA to solve issue

Bipolar vs CMOS

ES/9000 – 1990

- first series of S/390
- bipolar logic
- fast but very hot (biggest models water cooled)

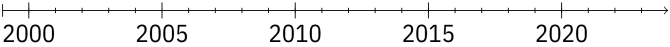
9672 – 1994

- CMOS
- Many optional features removed (e.g. vector instructions)
- Slower but cooler
    - g1 and g2 – mostly prototypes, used in very low end products
    - g3 and g4 – catching up
    - g5 and g6 – faster than bipolar

z/Architecture

Move to 64 bits in 2000.

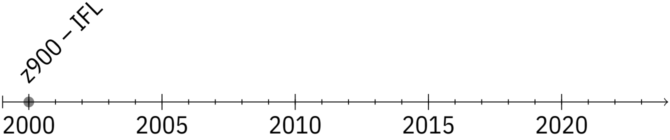
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

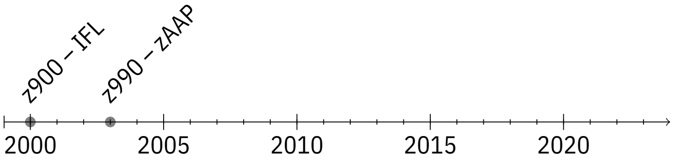
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

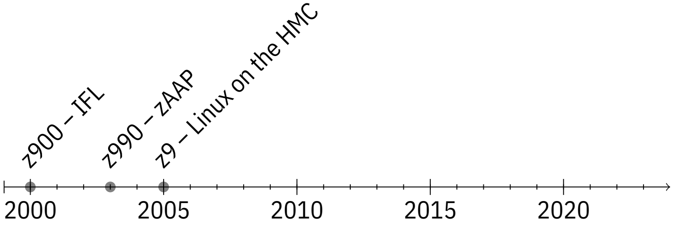
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

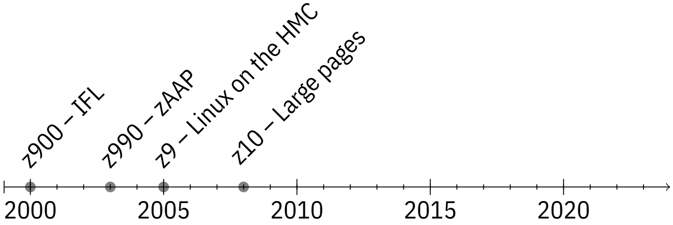
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

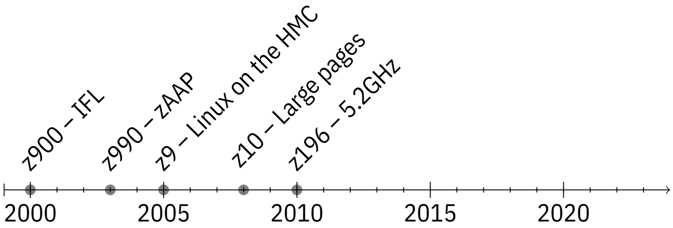
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

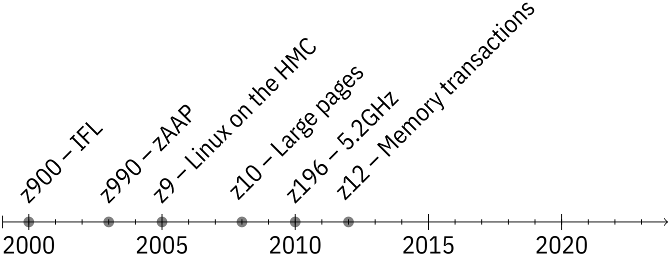
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

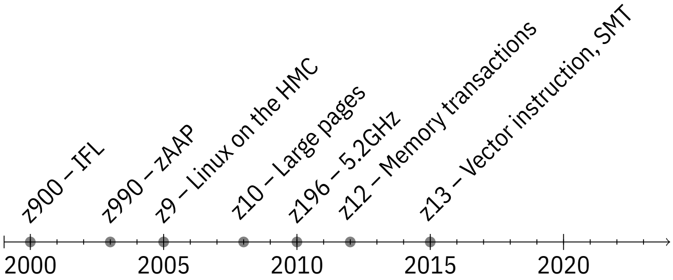
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z900 – IFL

z990 – zAAP

z9 – Linux on the HMC

z10 – Large pages

z196 – 5.2GHz

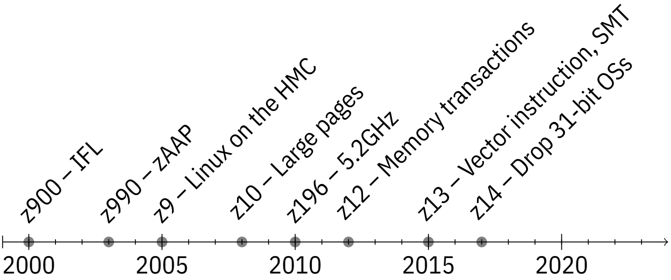
z12 – Memory transactions

z13 – Vector instruction, SMT

z/Architecture

Move to 64 bits in 2000.

- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z900 – IFL

z990 – zAAP

z9 – Linux on the HMC

z10 – Large pages

z196 – 5.2GHz

z12 – Memory transactions

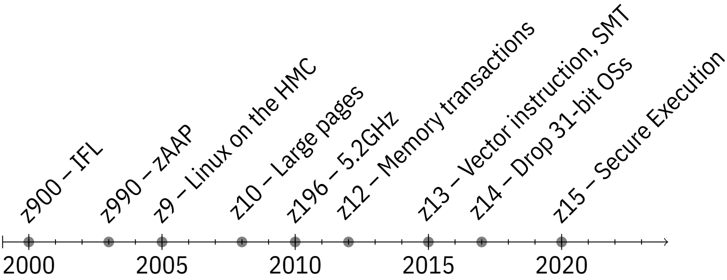
z13 – Vector instruction, SMT

z14 – Drop 31-bit OSs

z/Architecture

Move to 64 bits in 2000.

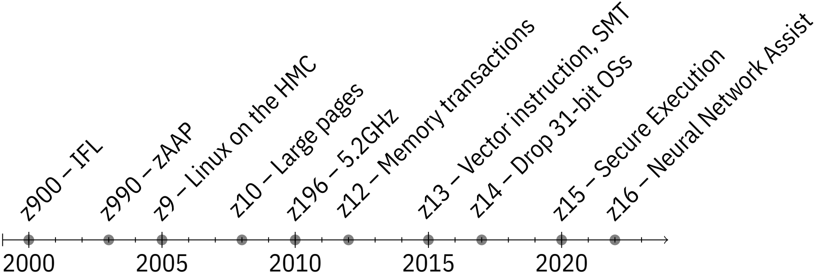
- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z/Architecture

Move to 64 bits in 2000.

- Registers extended to 64-bit, new instructions
- Backwards compatibility with 31-bit.
- Expanded storage not needed anymore to use more than 2 GiB.
- Page tables with variable number of levels, full 64-bit virtual address space
- Clone manufacturers did not keep up with the switch to 64 bits.



z900 – IFL

z990 – zAAP

z9 – Linux on the HMC

z10 – Large pages

z196 – 5.2GHz

z12 – Memory transactions

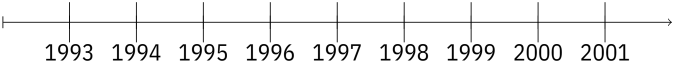
z13 – Vector instruction, SMT

z14 – Drop 31-bit OSs

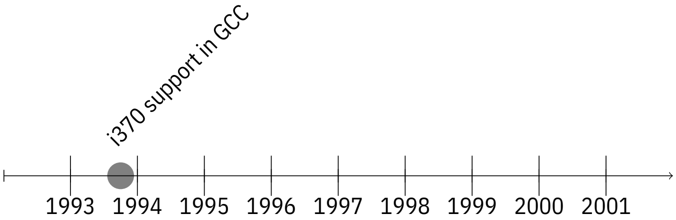
z15 – Secure Execution

z16 – Neural Network Assist

Linux for the mainframe - The compiler

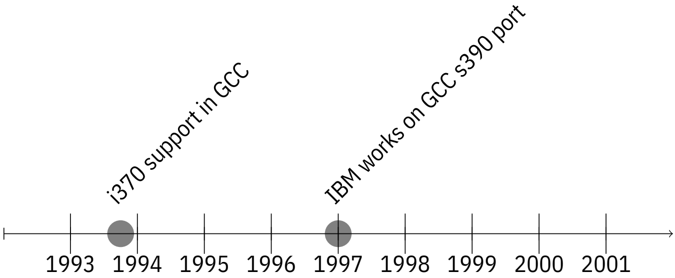


Linux for the mainframe - The compiler



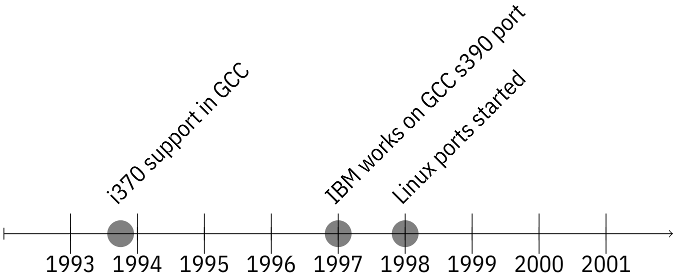
- October 1993: Code for the i370 backend appears in GCC repository
  - Developed outside IBM
  - Main objective: compile applications for MVS

Linux for the mainframe - The compiler



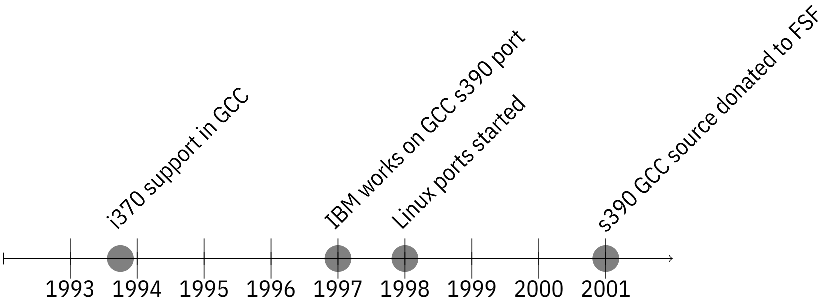
- 1997: IBM S/390 Firmware looking for a C compiler
    - Existing i370 backend for GCC is evaluated
    - S/390 port with less backward compatibility starts

Linux for the mainframe - The compiler



- 1998: Linux ports to the mainframe start
- Accelerates further development of compiler (i.e. ELF support)

Linux for the mainframe - The compiler



- 2001: GCC source code donated to Free Software Foundation

Linux for s390 - The IBM port `s390`

Linux for s390 - The IBM port `s390`

- 1998: IBM engineers prototype a Linux port to S/390 in their free time
- 18 December 1999: IBM releases source of Linux for S/390 on their FTP server

Linux for s390 - The IBM port s390

- 1998: IBM engineers prototype a Linux port to S/390 in their free time
- 18 December 1999: IBM releases source of Linux for S/390 on their FTP server
- January 2000: Code appears in Linux 2.2.14
- January 2000: First Linux Distro appears (Marist Linux by Marist College)

Linux for `s390` - `s390` and `i370`

- i370 also compatible with older machines
- i370 uses classical HLASM Assembly style, s390 does not
- Different toolchain
- i370 was less stable
- i370 developed by volunteers, s390 by IBM employees
- Linux i370 was abandoned when s390 port was published

Linux for s390 - Why it was interesting

Consolidation In 2000s datacenters consist of many physical boxes, with z/VM and Linux they can run on a single mainframe

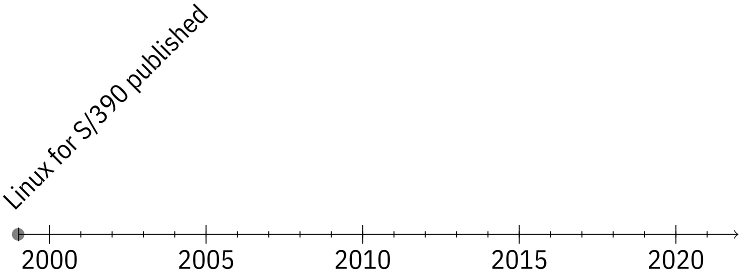
Java Java worked well on Linux

Unix Applications wanted a Unix-like OS

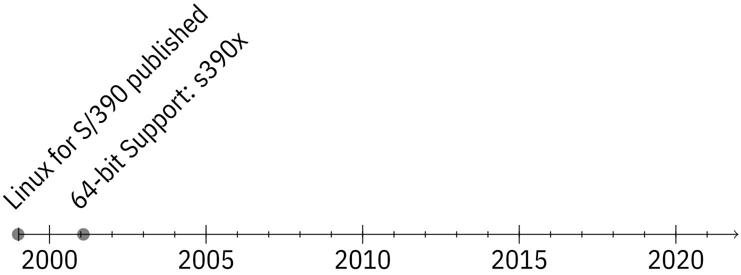
Save costs Comparatively cheap OS, later: cheaper CPUs for Linux (IFLs)

It was cool Many people just liked Linux

Linux for s390 - Evolution

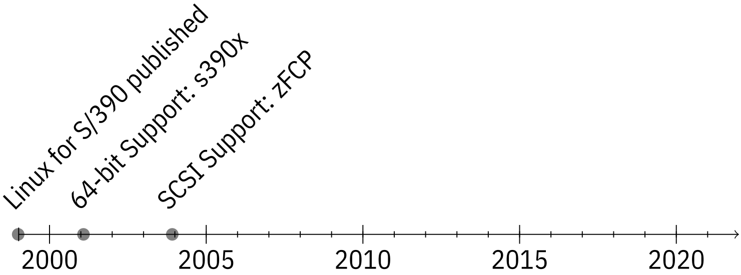


Linux for s390 - Evolution



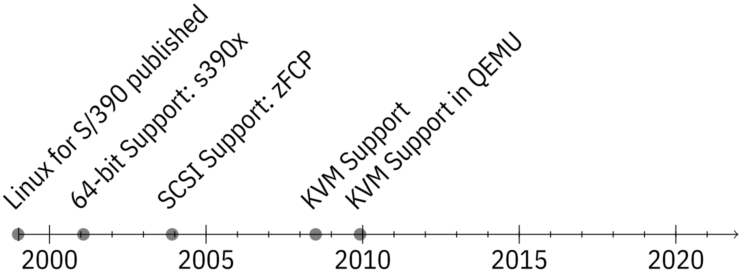
- 64-bit z/Architecture (s390x) supported since Linux 2.4.2 (Feb 2001)
  - 64-bit Kernels can run a 31-bit userspace fine (and still do so today)

Linux for s390 - Evolution



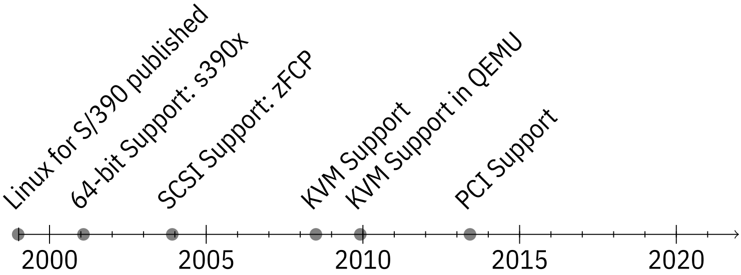
- SCSI Support (zFCP) since Linux 2.6.0 (Dec 2003)
  - Before that only DASD disks could be used, need special storage systems
  - Motivation: cheaper, more common storage

Linux for s390 - Evolution



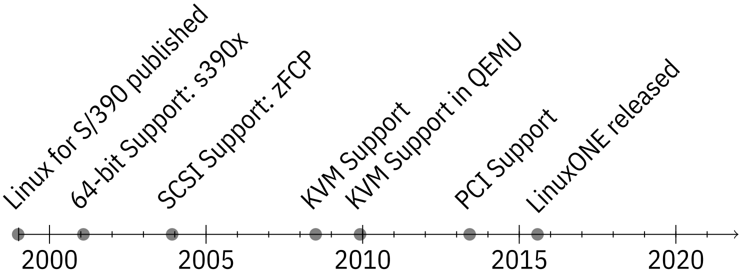
- KVM Support since Linux 2.6.26 (Jul 2008)
  - Initially with custom userspace `kuli`
  - Since December 2009, QEMU includes support for KVM
  - KVM Forum 2008: IBM shows 200 Linux VMs on a single Linux host

Linux for s390 - Evolution



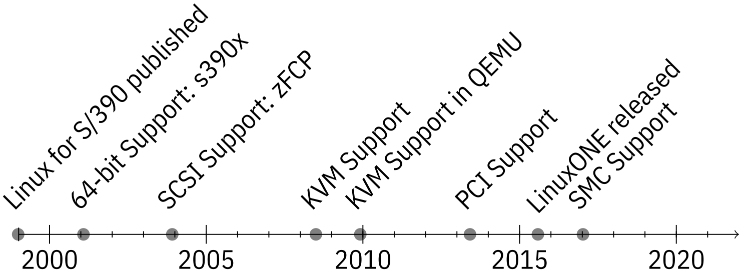
- PCI Support since Linux 3.10 (Jun 2013)
  - Standard interface to hardware (today: NVMe, network, HSMs, ...)
  - Can’t interface everything, you need custom hardware most of the time

Linux for s390 - Evolution



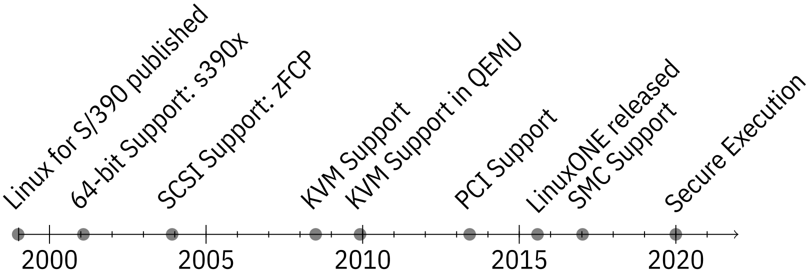
- LinuxONE: Linux-only system (Aug 2015)
  - Cheaper system which only runs Linux

Linux for s390 - Evolution



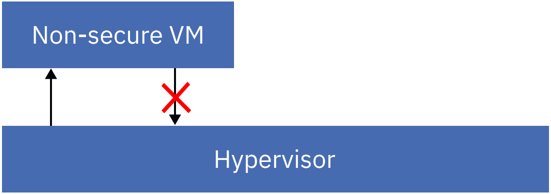
- SMC: Shared Memory Communications (Jan 2017)
  - Applications can establish a socket-like connection
  - Uses shared memory in the background
  - Works locally (i.e. between partitions) and remotely (between zSystems)

Linux for s390 - Evolution

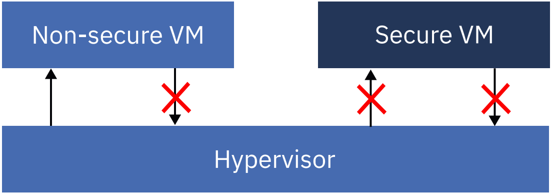


- Secure Execution: Confidential VMs (2020)

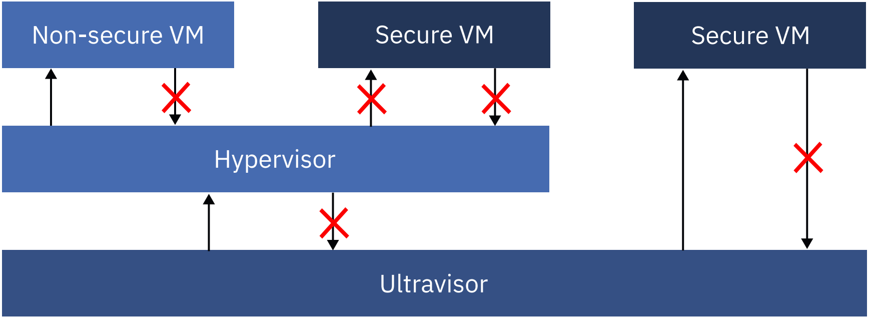
Linux for s390 - Secure Execution



Linux for s390 - Secure Execution



Linux for s390 - Secure Execution



References

Year	Reference	Title
1965	C24-3420-0	IBM System/360 Basic Programming Support and IBM Basic Operating System/360 Programming Systems Summary
1968	A22-6821-7	IBM System/360 Principles of Operation
1979	GA22-7070-0	IBM 4300 Processors Principles of Operation for ECPS:VSE Mode
1980	GA22-7074-0	Virtual-Machine Assist and Shadow-Table-Bypass Assist
1981	GA22-7000-7	IBM System/370 Principles of Operation
1984	SA22-7095-0	IBM System/370 Extended Architecture Interpretive Execution
1987	SA22-7085-1	IBM System/370 Extended Architecture Principles of Operation
1988	SA22-7200-0	IBM Enterprise Systems Architecture/370
2003	SA22-7201-08	Enterprise Systems Architecture/390 Principles of Operation
2022	SA22-7832-13	z/Architecture Principles of Operation

I want to play with this!

Hardware:

- Buy a mainframe (uhhhh.... yeah, right)
- Want to try your software on Big Endian? IBM LinuxONE Community Cloud: `https://linuxone.cloud.marist.edu/`
- zPDT
- Qemu for newer hardware
- Other emulators for older hardware

Software:

- OS/360
- DOS/360 and TOS/360
- CP-67/CMS and VM/370
- TSS/370

EOF