

**IBM Mainframe Operating Systems:  
Timeline and Brief Explanation  
For the IBM System/360 and Beyond**

**Dave Morton**

**Version 33 - September 2013**



# Contents

## Page

- 3 Short introduction and IBM Motto
- 4 Pictures
- 7 IBM operating systems which are known to run under Hercules
- 8 Brief history of the System/360
- 10 Summary of IBM mainframe operating systems
- 12 Operating System Progressions
- 13 Dates
- 15 Program Products statement (licensed products)
- 15 DOS/360 History
- 16 DOS, TOS, BOS, BPS
- 18 OS - PCP, MFT, VS1
- 20 OS - MVT, VS2, SVS, MVS, Z/OS
- 25 VM - Virtual Machine
- 26 TSS - Time Sharing System
- 26 ACP and TPF - High volume transaction processing
- 27 Miscellaneous Operating Systems:
  - OS/44 and PS/44.
  - IX/370 and AIX (IBM's version of Unix).
- 28 Non-IBM operating systems running on IBM mainframes:
  - MUSIC, MTS, Unix, Telpar, ORVYL, VP/CSS, UTS, PICK/370
- 29 Some components and features
- 39 Mainframe/PC Comparisons (for PC people NEW to mainframes)
- 63 Acknowledgements and Credits
- 63 Where to send comments

## INTRODUCTION

First introduced in 1965, these IBM mainframe systems are classic and timeless, and the latest versions are in wide use around the world today.

As such, they are a powerful testament to both the hardware and software architecture of the original IBM System/360.

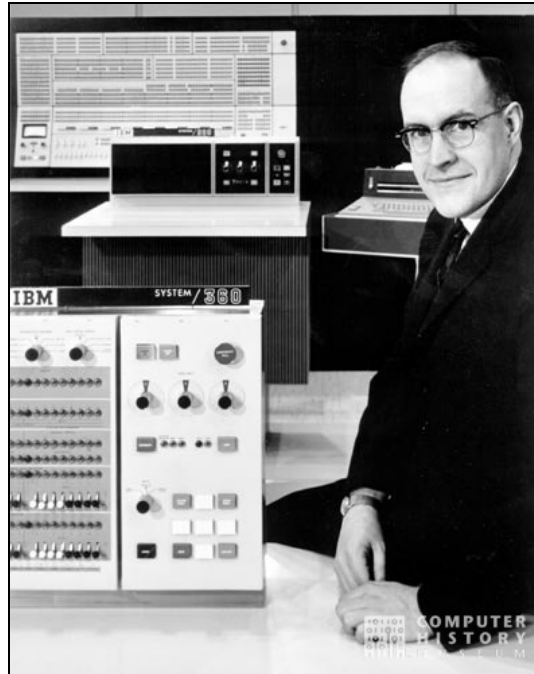


IBM Motto



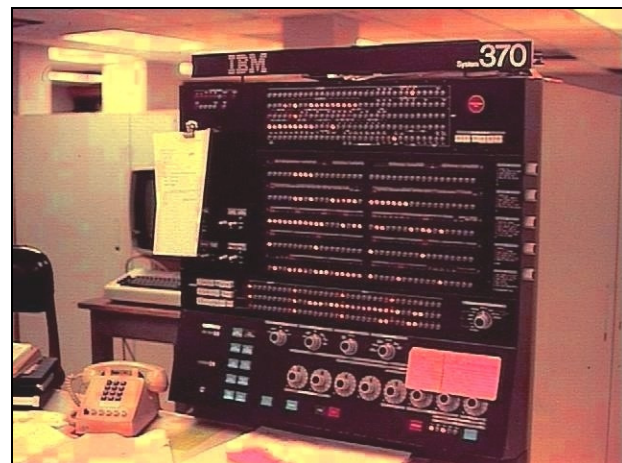
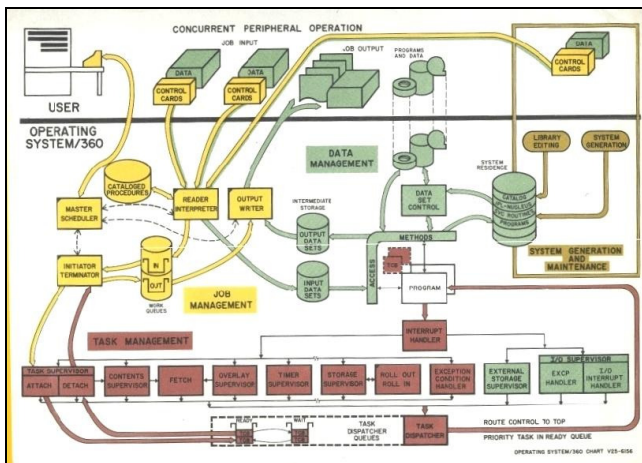
IBM Logo

## PICTURES



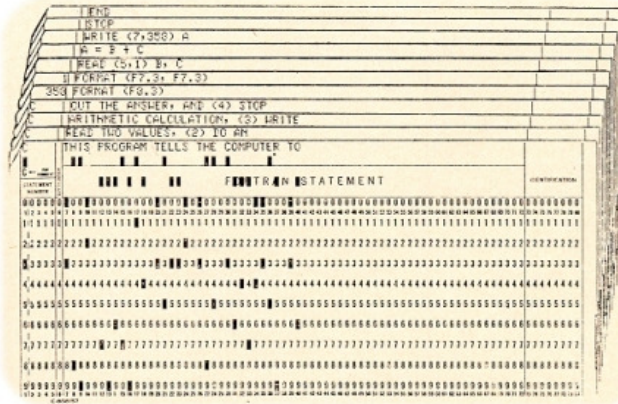
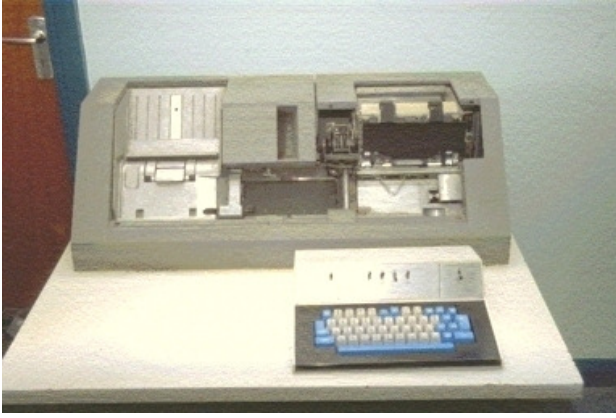
Left: One of the first ads for the System/360, circa 1964. The System/360 was first announced in April of 1964, and delivered to customers in 1965. The System/360 cost IBM \$5 billion to develop, which was second in cost only to the Apollo Moon program, during the decade of the 1960s.

Right: Dr. Frederick Brooks - the project manager of the S/360. He has won over 20 honors and awards, including the "National Medal of Technology" in 1985. (Gene Amdahl was the chief architect of the S/360.)



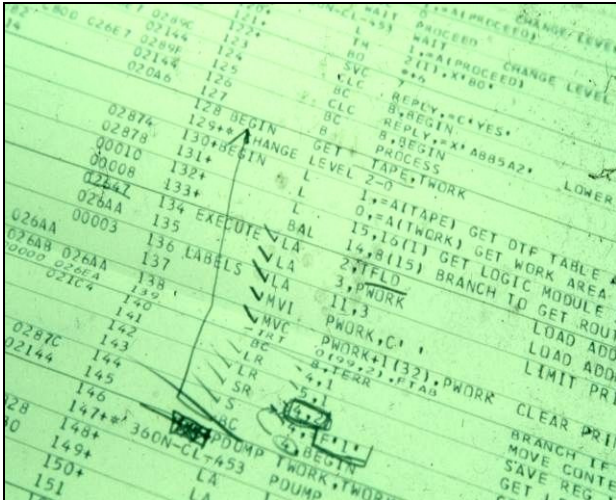
Left: OS/360: Concurrent Peripheral Operation, circa 1966 (multi-tasking, input and output queues, concurrent printing, etc).

Right: A System/370-155 operator's panel from the early/mid 1970s. It was one of the last IBM mainframes to have a panel with plentiful lights, dials, and switches.



Left: An 029 cardpunch (keypunch), circa 1960s/70s/80s. 029 and 129 keypunches were kept around for years, even though display terminals such as the 2260 became popular beginning in the late 1960s, and the 3270s in the 1970s.

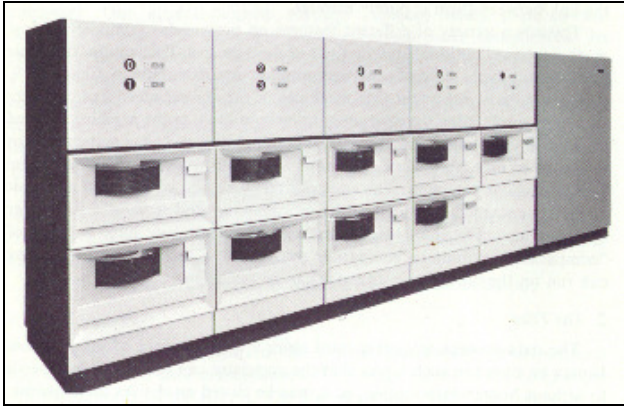
Right: A deck of punched cards for a FORTRAN program. FORTRAN was used mainly in colleges and research labs - not often in commercial shops.



Left: A DOS/360 program written in Assembler language. An Assembler listing from MFT/MVT/MVS, etc, would look nearly identical to the one shown. This program uses the older and more primitive "branch" mnemonics in part of the listing, such as "BC 8,BEGIN" instead of "BE BEGIN" - Branch if Equal, as it checks the reply for upper case 'YES' and lower case 'yes' before reading the tape using QSAM (GET/PUT), and specifying an optional work area named TWORK ("GET TAPE,TWORK").

COBOL was used for programming far more than Assembler, but every shop had (and still has) some Assembler programs for handling situations COBOL wasn't designed for. PL/1 and RPG were 2 other popular languages.

Right: A System/360 (possibly a model 67) with a 1052 console typewriter, and 2400 tape drives, circa late 1960s / early 1970s.



Left: 2314 disk drives (8 drives plus 1 offline spare) from the late 1960s/early 1970s (IBM ad). Sometimes referred to as "the pizza oven." The disk drive drawers could be pulled out by hand, and disk packs swapped. The I/O transfer rate for a drive was slow by today's standards, at about 312K bytes per second, but programs and files were smaller, then. (Disk drive speeds were dramatically improved in the 1970s). If 2 channels were attached to the control unit, any 2 drives could be reading or writing simultaneously. Each drive also had an associated "plug" shown on the top panel which could be removed and replaced (popped) to cause an interrupt if the system had stopped processing for unknown reason. This technique was rarely used, but there if you needed it.

Right: A 1403 printer on the left (early model), and a 2540 card reader/punch on the right. From the mid 1960s. The 1403, shown above, printed at 600 lines per minute. (The 1403-N1 printed at 1100 lines per minute).



Left: A System/360 with a 1052 console typewriter, 2400 tape drives, and 2314 disk drives in background, circa late 1960s / early 1970s (IBM ad).

Right: A 360/40 computer from 1969, used by a bank. This particular computer had 192K of core memory, and ran DOS/360. It ran 3 partitions (3 tasks): Background (BG) for batch jobs, and 2 partitions for online users (F1 and F2), handling over 30 teller terminals in one partition, and over 30 CRT terminals in the other, doing account inquiries, additions, and updates to master files.

## HISTORY

### Brief History of The System/360

Dr. Frederick Brooks from North Carolina was in charge of IBM's System/360 project. Brooks had previously been in charge of System Architecture for IBM, and had studied with Howard Aiken - one of the computer pioneers - at Harvard. Dr. Brooks is now a professor at the University of North Carolina, and has a homepage at:

<http://www.cs.unc.edu/~brooks/> .

He has won over 20 honors and awards, including the "National Medal of Technology" in 1985.

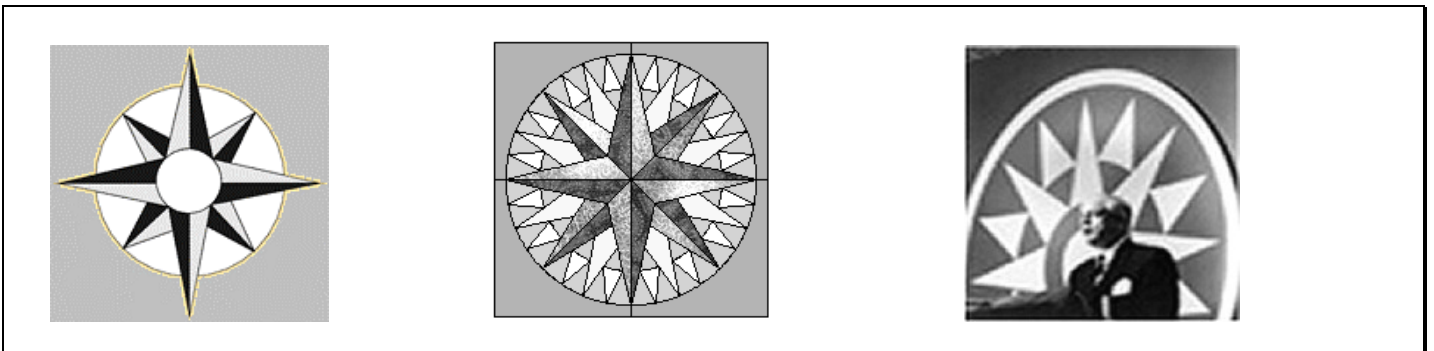
Dr. Gene Amdahl was the principal architect of the System/360, as well as the architect of some previous IBM computers.

The System/360 development utilized 60,000 people and cost IBM \$5 billion, which was second in cost only to the Apollo Moon program, during the decade of the 1960s.

The competition was hot on its heels with "1401" clones, and IBM had to borrow money to accelerate and complete the project. The System/360 announcement was made in April, 1964, and the first machines were delivered to customers in 1965.

The System/360 hardware (computer and peripherals) was ready by 1964. And I recall seeing a brochure for System/360 sitting on someone's desk in May, 1964, in a 1401 shop.

The title was "Introducing the System/360" with a drawing of a mariner's compass - an icon familiar to all who used the old System/360 manuals.



Style of the IBM System/360 Compass (a mariner's compass).

The official IBM System/360 compass logo is at the right behind Thomas Watson, Jr.

The compass was a symbol which meant "These are general-purpose computers usable by all industries, academic institutions, and government departments alike, and not tailored to one niche. They cover 360 degrees of the computing compass."

There was a second meaning for the term "System/360": A computing system for the 1960's (beginning in 1965 with the first delivery).

Later, a few models actually WERE tailored to particular niches, but they are branches on the System/360 tree, not part of the main trunk.

The term "System/370" means "IBM mainframe computing systems for the 1970s" - not "a compass with 370 degrees on it." Similarly, "System/390" refers to the 1990s. There was no "System/380", but there were IBM mainframe computers produced during the 1980's with model numbers such as 3081, 3083, etc.

Thus, the marketed names and meanings were:

<b>System/360</b>	Covers 360 degrees of the computing compass.
<b>System/370</b>	A computing system for the 1970s.
<b>Model 3081, etc</b>	A computing system for the 1980s.
<b>System/390</b>	A computing system for the 1990s.
<b>Zxxx</b>	A computing system for the 2000's.
<b>Z10</b>	A computing system for 2008...

"System/360" is often abbreviated "S/360", with "S/370" and "S/390" being abbreviations for 370 and 390 systems.

As good as the design of the original 360 was, it was the 360 Model 67, TSS, CP/67, and VM/CMS that moved IBM into the Virtual Storage and "online user" world with that model and subsequent 370 systems, with contributions from many people including IBM-Cambridge, MIT, Princeton University, Bell Labs, user groups, etc.

And let us not forget that the performance of the first releases of MVS was so bad that the MVS people in the SHARE user group adopted the Turkey as their symbol, with Turkey stickers stuck everywhere at their conferences - on buttons, on hats, on elevator doors - everywhere they could find a stickable surface, but later, adopted the Eagle as their symbol. The anti-TSO users had a button which said (in Latin), "Running TSO is like kicking a dead whale along the beach." Rough!!

Oddly enough, much of the MVS testing and development by IBM was done using a modified version of CP/67 and virtual 370's even before the 370 existed. This dependence upon VM (CP/67) probably saved VM, which had always had a "step-sister" image by much of IBM management. Even though early releases of MVS had performance issues, MVS IPL'd successfully on the first attempt on a real 370 with its VS features enabled, and it was all developed and tested on a VIRTUAL 370. MFT and MVT actually ran faster under VM/370 than on the real hardware.

The symbol of the VM group was the Teddy Bear since it was said to be better, warmer, and more user-friendly than MVS. But as MVS performance improved, the MVS folks retaliated with a sticker saying: "Real Men Don't Play With Teddy Bears".

Thanks to Melinda Varian for these VM anecdotes.

It could probably be said that the original design of the System/360, the foundational operating systems, and the design of the 360-67 and VM (CP/67) with its early use of virtual storage, are what brought us into the modern age of computing on both mainframes and desktops.

It's been a long and difficult road for the major operating systems, in one way or another. But now, at last, everything is perfect.

.....



## OS SUMMARY

### Summary of IBM Mainframe Operating Systems for the System/360 and Beyond

#### Generic Names

DOS, TOS, BOS, BPS (BPS - primarily a set of utilities and compilers)  
 OS, PCP, MFT, VS1 (and OS Standalone Utilities for any System/360)  
 MVT, VS2, SVS, MVS, z/OS  
 (CP/40), CP/67, VM, VM/CMS  
 TSS, ACP, TPF  
 OS/44, PS/44  
 IX, AIX

#### DOS/360 and Related Operating Systems and Utilities:

DOS - Disk Operating System. 1965. [disk-based]  
 TOS - Tape Operating System. 1965. [tape-based]  
 BOS - Basic Operating System. 1965. [card/tape/disk-based]  
 BPS - Basic Programming Support. 1965. [primarily utilities and compilers]  
 DOS/VS - DOS/Virtual Storage. 1970s.  
 DOS/VSE - DOS/Virtual Storage Extended.  
 SSX/VSE - Small System executive (supervisor) for VSE.  
 DOS/VSE/AF - DOS/VSE/Advanced Functions.  
 VSE/SP - VSE/System Product. 1984.  
 VSE/ESA - VSE/Enterprise Systems Architecture. 1990.  
 z/VSE - VSE for Z-Series mainframes (24-bit and 31-bit modes only).

#### OS/360 and Related Operating Systems and Utilities:

OS - PCP, MFT, and MVT operating systems. 1966.  
 OS Util - OS Standalone Utilities (not the utilities of the OS). 1966.  
 OS/PCP - Primary Control Program. One Region. March, 1966.  
 OS/MFT - Multiprogramming with a Fixed number of Tasks. 1966.  
 OS/MFT-II - Usually just called "MFT" without the "II". 196x.  
 OS/VS1 - Virtual Storage 1, from MFT. 1972.  
 OS/VS1/BSEPP - VS1 Basic Systems Extension Program Product.  
 OS/MVT - Multiprogramming with a Variable number of Tasks. 1966.  
 OS/VS2 R1 - Virtual Storage 2 Release 1 (later known as SVS) from MVT. 1972.  
 SVS - Rename of OS/VS2 R1. Single Virtual Storage space.  
 OS/VS2 R2 - Virtual Storage 2 Release 2 (later known as MVS). 1974.  
 MVS - Rename of OS/VS2 R2 and above. Multiple Virtual Storage spaces.  
 MVS/SE1 - System Extensions Release 1. 1978. 5740-XE1.  
 MVS/SE2 - System Extensions Release 2. 1979. 5740-XE1.  
 MVS/SP1 - MVS System Product Version 1. June, 1980.  
 MVS/SP2 - MVS System Product Version 2. June, 1980.  
 MVS/SP3 - MVS System Product Version 3. November, 1980.  
 MVS/370 - Rename of MVS, MVS/SE, and MVS/SP when MVS/XA was announced.  
 MVS/XA - Extended Architecture with 31-bit addressing. 1983.  
 MVS/ESA - Enterprise Systems Architecture.  
 OS/390 - MVS/SP6. 1996.  
 z/OS - MVS for Z-Series mainframes. 24/31/64-bit modes. 2001.

#### VM Operating System (Virtual Machine):

(CP/40) - Control Program for System/360 Model 40. 1966.  
 CP/40 was an IBM internal product. Not released to customers.  
 CP/67 - Control Program for the System/360 model 67. 1967.  
 CMS - Cambridge Monitor System, then Conversational Monitor System.  
 VM - Virtual Machine.  
 VM/370 - Virtual Machine for System/370.  
 VM/370 R6 BSEPP - VM Basic System Extension Program Product.  
 VM/370 R6 SEPP - VM Systems Extension Program Product.  
 VM/SP - VM System Product.  
 VM/Entry - VM/SP R3.1 - Entry level system.  
 VM/IS - VM/SP R6.n - Information Systems.

VM/SP HPO - High Performance Option - early 1980s  
 VM/ESA/370 - For 370s. Non-XA version. 1990.  
 VM/XA MA - XA Migration Aid (a version of VM with 31-bit addressing)  
 VM/XA SF - Extended Architecture / System Facility.  
 VM/XA SP - System Product. Approx 1987.  
 VM/ESA - Enterprise Systems Architecture. 1990.  
 z/VM - VM for Z-Series mainframes. 24/31/64-bit modes.

#### Other IBM Operating Systems:

TSS = The acronym for "Time Sharing System". See below.  
 TSS/360 - TSS for System/360 (ran on model 360/67 only).  
 TSS/370 - TSS for System/370 with Virtual Storage enabled.  
 ACP - Airline Control Program.  
 TPF - Transaction Processing Facility.  
 OS/44 - An operating system for the model 360/44 (scientific computer).  
 PS/44 - An operating system for the model 360/44 (scientific computer).  
 IX - Interactive Executive (IBM's Version of UNIX).  
 AIX - Advanced Interactive Executive (IBM's Version of UNIX).  
 IX/370 - Interactive Executive for 370s.  
 AIX/370 - Advanced Interactive Executive for 370s.  
 AIX/ESA - AIX for Enterprise Systems Architecture - 1991.  
 RAX - Predecessor of MUSIC. Enhanced by McGill University to create MUSIC.  
 MUSIC/SP - Adopted by IBM from McGill as a System Product.  
 Z/Linux - Linux on z-Series mainframes.  
 Z/TPF - TPF on z-Series mainframes.

#### Non-IBM Operating Systems:

MUSIC - McGill University System for Interactive Computing. 1975.  
 MTS - Michigan Terminal System.  
 UNIX for System/370 - Bell Labs, 1979/1980.  
 TELPAR - For the design and testing of electronics. Late 1960s.  
 ORVYL from Stanford University - <http://en.wikipedia.org/wiki/ORVYL>  
 VP/CSS from National CSS - <http://en.wikipedia.org/wiki/VP/CSS>  
 UTS from Amdahl - [http://en.wikipedia.org/wiki/UTS \(Mainframe UNIX\)](http://en.wikipedia.org/wiki/UTS_(Mainframe_UNIX))  
 PICK/370 from SMI and Ultimate - [http://en.wikipedia.org/wiki/Pick operating system](http://en.wikipedia.org/wiki/Pick_operating_system)

#### Dual Development Operating Systems:

MUSIC/SP - Multi-User System for Interactive Computing.  
 (IBM and McGill University). IBM System Product - 1985.

## PROGRESSION OF MAJOR IBM OPERATING SYSTEMS

### Schematic Diagram of IBM Mainframe Operating System Progressions

Key: Green = Operating System is in use today in advanced versions.

#### DOS/360 - Disk Operating System:

DOS --> DOS/VS --> DOS/VSE --> (SSX/VSE for 43xx's) -->  
 DOS/VSE/AF --> VSE/SP --> VSE/ESA --> z/VSE

#### TOS/360 - Tape Operating System:

TOS --> discontinued (not widely used)

#### BOS/360 - Basic Operating System:

BOS --> discontinued (small, card/tape/disk OS - not widely used)

#### BPS/360 - Basic Programming Support:

BPS --> discontinued (primarily utilities and compilers for DOS/TOS/BOS)

#### OS/360 (family of 3 operating systems): PCP, MFT, and MVT.

OS/PCP --> discontinued (used by IBM and some customers)

OS/MFT --> OS/MFT-II --> OS/VS1 --> discontinued

OS/MVT --> OS/VS2 R1 (SVS) --> OS/VS2 R2 (MVS) -->

MVS/SE1 --> MVS/SE2 --> MVS/SP -->

MVS/XA --> MVS/ESA --> OS/390 --> z/OS

OS Standalone Utilities --> discontinued (used by IBM and some customers)

#### VM - Virtual Machine:

CP/67 (on a 360/67) --> VM/370 --> VM/370 BSEPP --> VM/370 SEPP -->

VM/SP --> VM/SP/HPO --> discontinued

VM/SP --> VM/ESA/370 --> VM/XA MA --> discontinued

VM/ESA/370 --> VM/XA SF --> VM/XA SP --> VM/ESA --> z/VM

#### ACP - Airline Control Program / Transaction Processing Facility:

(Online transaction processing for airlines, hotels, etc).

ACP --> TPF --> z/TPF

#### TSS - Time Sharing System:

(Online development system).

TSS/360 (on a 360/67) --> TSS/370 --> discontinued

(replaced by VM/CMS and TSO)

#### OS/44 and PS/44:

(low-cost scientific/academic system on 360/44).

OS/44 and PS/44 --> discontinued

Not listed: z/Linux, MUSIC/SP.

#### Hercules notes:

\* Model 360/67 used by CP/67 and TSS/360 is not supported by Hercules.

\* Model 360/44 used by OS/44 and PS/44 is not supported by Hercules.

\* Model 360/20 (a mini-computer) is not supported by Hercules.

The 360/20 could run 3 OS's: DPS (Disk Processing System),

TPS (Tape Processing System), and CPS (Card Processing System).

.....

## DATES

The System/360 was first announced by IBM on April 7, 1964.  
 Deliveries to customers began about 1 year later in 1965.  
 First software delivered: DOS/360, TOS/360, BOS/360, and BPS/360.

### Dates of Delivery to Customers

```

-----
Year
Del.  Name
-----
1965  System/360  Hardware with DOS/360, etc - not OS/360.

1965  DOS/360, TOS/360, BOS/360, BPS/360 - For smaller systems.

1966  OS/360       PCP, MFT, and MVT - For larger systems.

1967  CP/67        Early version of VM (Virtual Machine). For S/360-67.

1967  TSS/360      Early version of time sharing with virtual storage.

1972  VS1          Virtual Storage version of MFT + enhancements.

1972  VS2 R1 (SVS) Virtual Storage version of MVT + enhancements.

1974  VS2 R2 (MVS) VS2 Release 2.
                    16-megabyte addressing (24 bits) per address space.

1974  MVS          Same as "VS2 Release 2". Synonym - not a release.
                    Easier to say than "VS2 Release 2".

1974  (MVS/370)    Refers to MVS before MVS/XA.
                    MVS/370 means "24-bit MVS". MVS/XA = 31-bit MVS.
                    Not an official IBM term, but explanatory.
                    MVS/XA was released in 1983, 9 years later.

1974  MVT R21.8F   2nd to last release of MVT. Many fixes applied.

1977  MVS R3.8     MVS Release 3.8 (OS/VS2 R3.8)

1978  MVT R21.x    The last release of MVT.
                    For use on 3031, 3032, and 3033 systems only
                    (VS-capable systems), for customers not wanting
                    to convert to VS2 yet.

1978  MVS/SE1     MVS Systems Extensions Release 1

1979  MVS/SE2     MVS Systems Extensions Release 2

1980  MVS/SP1     MVS/SP Version 1 (June, 1980). SP = System Product.
                    Later renamed to "MVS/SP1.1".

1980  MVS/SP2     MVS/SP Version 2 (June, 1980).
                    Later renamed to "MVS/SP1.2".

1980  MVS/SP3     MVS/SP Version 3 (November, 1980).
                    Later renamed to "MVS/SP1.3".

1981  MVS R3.8J   MVS Release 3.8J.
                    16-megabyte addressing per region (24 bits).
                    3.8J precedes functionality of MVS/SE,
                    MVS/SP, MVS/XA, etc.
  
```

Note: MVS 3.8 could run 2 CPU's quite well, but did not utilize more than 2 CPU's to any satisfactory degree. MVS/XA (the next version) utilized more than 2 CPU's well.

1983	MVS/XA	MVS/SP Release 2 (MVS/SP2.0). XA = Extended Architecture. Extended Architecture with greater addressing. 2-gigabyte addressing (31 bits) per address space rather than 16-megabyte addressing (24 bits). Includes everything in MVS/SP 1.3 plus enhancements. XA utilized more than 2 CPU's well.
1985	MVS R3.8J + maint	MVS Release 3.8J with maintenance to level 8505. 16-megabyte addressing per region (24 bits). 3.8J precedes functionality of MVS/SE, MVS/SP, MVS/XA, etc, despite the later date. Not at the MVS/XA level. Functionality level = 3.8J, 1981. Maintenance level = 1985.
1988	MVS/ESA	MVS/SP Release 3. MVS continued. ESA = Extended Systems Architecture.
19xx	MVS/ESA	MVS/SP Release 4. MVS continued. Base SCP 3.8 dropped.
1995	OS/390	MVS/SP Release 6. MVS continued.
2001	z/OS	MVS / OS/390 continued. For Z-series 64-bit mainframes.

Note 1: SCP means "System Control Program", IE, "The Operating System". Successive Program Product versions of MVS were "installed over" the Base SCP until MVS/SP4, which is a version of OS/390.

Note 2: These acronyms, versions, releases, SCP Bases and dates can be a tad confusing...

Note 3: Hercules note:

"MVS/3.8J with maintenance" (1985) is mentioned because this is the release of MVS supplied in the Hercules TK3 Turnkey MVS from Volker Bandke, via Jay Maynard, via IBM. It is "plain MVS" at the 3.8 level - not a Program Product such as SE, SP, or XA, but it works very well, thank you! It is in the Public Domain, and represents more than 15 years of "OS/MVT" development, from MVT to MVS, from 1966 to 1981.

MVS/3.8J Functionality Level: 1981.

MVS/3.8J Maintenance Level: 1985.

Note: MVS 3.8 could run 2 CPU's quite well, but did not utilize more than 2 CPU's to any satisfactory degree. MVS/XA (the next MVS version) utilized more than 2 CPU's well.

User of Hercules (the IBM mainframe emulator) will probably not see any performance improvement specifying more than 2 CPU's in their computer hardware definition with any version of MVS 3.8. With the high speeds of today's PC's (2012, etc), CPU performance should be more than satisfactory.

## PROGRAM PRODUCTS / SYSTEM PRODUCTS / LICENSED PRODUCTS

### (Licensed Products)

The terms "Program Product", "System Product" and "Licensed Product" mean:

"A product which may be licensed from IBM for a fee."

These Licensed Products include: SPF (ISPF), SDSF, CICS, IMS, DB2, RACF, RMF, VSE/ESA, MVS/SPx, OS/390, z/OS, VM/ESA, current versions of SORT (DFSORT), High Level Assembler (HLA), COBOL/370, and many others.

They must be obtained from IBM, and a license is required from IBM to run them. They are expensive.

Hercules note: A few substitute programs have been written by volunteers to provide partial functionality. See the Hercules FAQ documentation.

## DOS/360 HISTORY

### DOS/360 History - The First Operating System for the System/360

DOS/360 was developed out of necessity.

It was smaller and simpler than OS/360, and was written so that an operating system would be available for the computer, since OS/360 was taking longer to complete than had been planned for.

DOS/360 was released in 1965 before OS/360, along with TOS/360, BOS/360, and BPS/360. (TOS was the "tape" version of DOS. BOS was for smaller 360 computers, and BPS was a set of utilities.)

DOS was more basic and had fewer automatic functions than OS/MFT and OS/MVT. For example, it had no spooling capabilities and lacked a system catalog. There was very little in the way of "device independence" for programmers: If you specified a printer as an output device in your program, it could not be routed at run-time to a disk or tape drive (unless you used the DTFDI macro in Assembler for "Device Independence", but few people used it, I believe). Under OS, routing a printer listing to tape or disk is just a matter of changing a JCL statement before submitting the Job. But DOS did the job. It worked, and it sold computers.

An add-on component called POWER, introduced for DOS in the late 1960s, added spooling capabilities. Today's VSAM files are cataloged in a VSAM catalog, giving VSE (the new DOS) a partial "system" catalog - at least for VSAM files.

It was a very popular OS, and continues to this day in its far more advanced versions. However, if you've used MVS, it's difficult to drop back to the DOS-based versions of today (other than for "hobby" purposes). The POWER control cards are rather strange and the terminology is different from MVS JCL. If you've never used MVS, or you're just more comfortable with it, it's great.

IBM eventually changed the name "DOS" to avoid confusion with its DOS for the PC, so the mainframe "DOS" became known as "VSE", while DOS for the PC became "PC/DOS".

DOS for the PC was officially called:

"IBM Personal Computer Disk Operating System" for the first 3 releases. It eventually became "PC/DOS" (IBM's version) and "MS/DOS" (Microsoft's version).

## DOS, TOS, BOS, BPS

### DOS/360, Variants and Progressions

DOS - Disk Operating System. 1965. [disk-based]

The operating system existed on disk only.

Also included: Assembler, COBOL, RPG, PL/1, and FORTRAN languages.

DOS/VS - DOS/Virtual Storage. 1970s.

DOS/VSE - DOS/Virtual Storage Extended

SSX/VSE - Small System eXecutive (OS) for VSE.  
Pre-genned VSE for 4331, 4341 and 4361 systems.  
No direct use of JCL. Installs in 1 to 2 hours.

DOS/VSE/AF - DOS/VSE/Advanced Functions

VSE/SP - VSE/System Product. 1984.

VSE/ESA - VSE/Enterprise Systems Architecture. 1990.  
Can be run in 370 mode or ESA mode.

z/VSE - VSE for Z-Series mainframes (24/31-bit modes only - not 64-bit).  
20xx.

TOS - Tape Operating System. 1965. [tape-based]

The operating system existed on tape only.

The "tape" version of DOS.

Almost identical to DOS except for the location of the operating system.  
Slow. Phased out during DOS's lifetime.

More of an "entry level" OS which could be sold to potential customers who couldn't afford disk drives.

Later, they could upgrade their systems with disk drives and use DOS instead of TOS.

BOS - Basic Operating System. 1965. [card/tape/disk-based]

A small, operating system similar to DOS, primarily intended for System/360 Model 30 computers with 8K of memory and above.

Model 30 memory sizes:

8K, 16K, 24K, 32K, 64K (models C30, D30, DC30, E30, F30).

Came in several versions. For example, a "16K Tape" version existed in 1965 for use with 4 tape drives.

Assembler and RPG languages included.

Entry-level system.

Amazing fact: Had a Spooling system for queued printing. DOS did not.

BPS - Basic Programming Support. 1965. [card-based]

A set of utilities and compilers.

The programs and their mini-OS existed on cards only.

Provided basic, standalone utilities, such as initializing tapes and disks, for DOS/TOS/BOS, as well as a card-based assembler and linkage editor.

An alternative to using BPS was to use OS standalone utilities for basic functions such as initializing tapes, etc.



## PCP, MFT, VS1

### OS/360, Variants and Progressions.

-----  
 (OS/360 family = OS/PCP, OS/MFT, and OS/MVT)

The type of OS to generate (PCP/MFT/MVT) was specified at SYSGEN time.

Also included: Assembler, COBOL, RPG, PL/1, and FORTRAN languages.

OS/PCP - Primary Control Program. March, 1966.  
 First member of the OS/360 family to be delivered to customers, although the customers normally later replaced it with MFT or MVT when they became available.  
 Could run only 1 program at a time.  
 No program "Partitions", just 1 big area for 1 program.  
 No spooling capabilities: Card readers and printers were allocated by device, such as UNIT=00C, UNIT=00E, etc.  
 Used by IBM development to create the tools for the other, more complicated operating systems, and used by some customers.

OS/MFT - Multiprogramming with a Fixed number of Tasks. 1966.  
 Used partitions, changeable in size and number by the operator.  
 1 program per partition.  
 Max number of partitions: Fixed at SYSGEN time.  
 Partitions were called "P0, P1, P2...".

The number of tasks was not really "fixed" - just the max number of partitions, and that could be changed by doing a SYSGEN. The MFT name is a misnomer.

MFT could be called:

"Multiprogramming with a semi-Fixed number of PARTITIONS, although you can change them".

Partition sizes could be changed by the operator.

Maximum number of \*user\* partitions gennable: 15.  
 Readers and writers used their own, additional partitions.  
 Max Readers: 3.  
 Max Writers: 36.  
 Max number of Initiators: 15.  
 Max number of concurrent Jobs: 15.  
 Max Partitions: 52.  
 Max Tasks: 255.

Storage protection (writing to memory): Yes.  
 Fetch protection (reading from memory): No.  
 Storage Protect Keys: 15. (hardware keys)

Available options:

Time slicing: Yes.  
 Checkpoint/restart: Yes.  
 SMF: Yes.  
 TSO: No.  
 Can use HASP: Yes.  
 Multiprocessing: No.

Minimum memory: 128K.

(The above features are a combination of MFT and a later release called MFT-II).

Within a partition, 1 main program could be running. The "main program" could be a Job or a "Started Task". With multi-tasking, (introduced in MFT-II), additional "tasks" could be attached by a main program using the ATTACH macro.

Thus, the number of running main programs, plus their tasks [if any], could theoretically be quite large. MVT and the other OS's (including DOS) also supported multi-tasking.

A Started Task is a program started by the operator by using the "START" command for a PROC name, instead of being submitted to the system in the "normal" fashion as a Job (although the system generates a Job card for it, anyway). For example: "S MF1" (START MF1) means "Execute the PROC named MF1".

IBM internally referred to the Readers, Writers and Initiators as "JES", but the term wasn't commonly used publicly until VS1 and VS2 were announced. Then we found out that the subsystem was called JES.

The term "JES" was not used in MFT or MVT manuals. JES later was renamed JES1 in VS1.

HASP later was renamed JES2 in VS2.

The original JES was retained in OS/VS2 R2, but JES2 or JES3 were layered on top of it, for the most part.

MFT and MVT also used "RES", but the term was not used in MFT and MVT manuals.

Note that PCP, MFT, MVT, and the other older OS's are no longer distributed or supported by IBM, and are no longer used commercially as far as I know.

MFT and MVT were discontinued by IBM in the 1970s, and were replaced by the "Virtual" operating systems OS/VS1 for MFT and OS/VS2 for MVT. Similarly, DOS was replaced by DOS/VS.

OS/MFT-II - Enhanced MFT.

Usually just called "MFT" without the "II".

OS/VS1 - Virtual Storage 1. 1972.  
16 meg max addressable memory per system.

Maximum number of user partitions gennable: 15.  
Readers and writers used their own, additional partitions.

Max Readers: No limit.

Max Writers: No limit.

Max number of Initiators: 15.

Max number of concurrent Jobs: 15.

Max Partitions: No limit.

Can use HASP: I don't think so.

Spooling/Readers/Writers/Initiators: JES.

Storage protection (writing to memory): Yes.

Fetch protection (reading from memory): Yes.

Minimum memory: 160K.

OS/VS1/BSEPP - 3380 DASD support

## MVT, VS2/SVS/MVS, z/OS

OS/MVT - Multiprogramming with a Variable number of Tasks. 1966.  
Used variable-sized program execution areas called "Regions" (the analog to MFT's "Partitions"), specifiable in size by JCL at run time. Example: REGION=48K.

Maximum number of user Regions gennable: 15.  
Readers and writers used their own, additional Regions.  
Max Readers: No limit.  
Max Writers: No limit.  
Max Regions: No limit.  
Max number of Initiators: 15.  
Max number of concurrent Jobs: 15.

Storage protection (writing to memory): Yes.  
Fetch protection (reading from memory): No.  
Storage Protect Keys: 15. (hardware keys)

### Available options:

Time slicing: Yes.  
Checkpoint/restart: Yes.  
SMF: Yes.  
TSO: Yes.  
Can use HASP: Yes.  
Multiprocessing, loosely coupled: Yes, with ASP.  
Multiprocessing, tightly coupled: Yes, with Model MP65.  
Special version of MVT needed for Model MP65  
multiprocessing. The 360/MP65 had 2 CPU's.  
Multiprocessing, remote: Yes, with ASP and RJP.

Minimum memory: 256K.

Highest MVT release: 21.8F - 1974.

A special modification of 21.8F was released in 1978 for potential 303x customers, who needed the greater capacity of a 303x machine, running it without the DAT box enabled, who wanted to use MVT rather than convert to VS.  
Functionality level = 1974.

MVT offered TSO as an optional component; RO/RI (Roll Out/Roll In) was used for swapping TSO users into and out of memory, as needed, and was called "Swapping" even before the days of virtual storage.  
TSO access method: TCAM.

"Fetch protection" wasn't added until VS1 and VS2, R1. This hardware security feature prevents an application program from reading the contents of another Region.

The number of active Regions and the size of each Region were variable, making it more flexible than MFT. However, Regions were often allocated too large to ensure enough memory would be present, wasting memory. Also, memory fragmentation became a problem, which is one the many reasons why IBM developed "virtual storage" for its next line of OS's.

MVT is the basis for today's MVS, OS/390 and z/OS operating systems. It can run on 370 hardware, although it doesn't use the virtual storage features or the new PSW format.

\* Hercules note: OS/MVT from Jay Maynard is MVT 21.8F.

OS/VS2 R1 - Virtual Storage 2 Release 1. (SVS). 1972.

16 meg max addressable memory per system.

SVS

Single Virtual Storage space.

| Later known as SVS.

| (Originally called SVM - "Single Virtual Memory".)

| Max number of Initiators: 63.

| JES: Yes.

| JES2: No.

| JES3: No.

| Can use HASP: Yes.

| Storage protection (writing to memory): Yes.

| Fetch protection (reading from memory): Yes.

| Minimum memory: 384K.

V R1 ---> SVS Rename from "OS/VS2 Release 1".  
Single Virtual Storage space.  
SVS highest release: 1.7

OS/VS2 R2 - Virtual Storage 2 Release 2. (MVS). 1974.

Multiple Virtual Storage spaces.

MVS

16 meg max addressable memory per Address Space

(in other words, 16 MB "per Region", using MVT terms).

"OS/VS2 R2 And Above" was later known as "MVS".

Also later renamed to "MVS/370" for versions before MVS/XA.

Max number of Initiators: No limit.

Spooling/Rdrs/Wtrs/Initiators: JES2 or JES3.

JES: Yes. (still embedded in OS)

JES2: Yes. (derivative of HASP)

JES3: Yes. (derivative of ASP)

Specify JES2 or JES3 at SYSGEN.

Can use HASP: Doubtful (unnecessary with JES2).

JES2 was much more commonly used than JES3.

Note: One Herculean says he has run MVS without

JES2 or JES3 (although MVS abended due to a

needed EC change).

Master Catalog had to be a VSAM catalog, but CVOLS from  
the old OS Catalog method (CVOL Catalog) were also supported.

24 bit, virtual. Minimum memory required: 768K.

Some software typically used by customers:

JES2 or JES3 (usually JES2),

CVOL Catalogs, VSAM, VSAM catalogs, BTAM, TCAM,

VTAM, TSO, SPF, BSAM, QSAM, BDAM, BPAM, SMF, RMF,

SMP. Some ISAM, but eventually replaced with VSAM.

Programmer online access: SPF (and TSO).

Online applications: CICS, usually.

Database applications: IMS, usually.

System measurement: RMF, usually.

V R2 ---> MVS Rename from "OS/VS2 R2 and above".  
Multiple Virtual Storage Spaces.  
It was easier to say "MVS" than  
"OS/VS2 Release 2 and above".  
Internal name: AOS/2 (or AOS).  
Plain MVS highest release: 3.8J.

Approx 1981 + later maint.

- \* MVS 3.8J from Jay Maynard:
- \* Service level 8202 (1982) -
- \* Maint for entire OS. And...
- \* Service level 8505 (1985) -
- \* Maint for base OS + SMP4.
- \* SMP version -
- \* Starter system: 4.12
- \* smp4.aws tape: 4.22
- \* In Files area: 4.45 (SMP4.XMIT)
- \* (Use the SMP 4.45 version.
- \* Blocked for 3330. Load it to a
- \* workpack with dasdload. Then
- \* IEBCOPY it to SYS1.LINKLIB.
- \* Cum tape for PTFs -
- \* cum0003.aws.gz in Files area.
- \* Assembler XF.
- \* No SPF or SDSF (licensed products).

- MVS/SE1 - System Extensions Release 1. 1978. 5740-XE1.  
Program Product.  
Hardware performance improvements.  
Some OS functions placed in microcode.  
TSO logical swapping added (the TSO user would  
Remain in memory but be marked as being "logically  
swapped out").
- MVS/SE2 - System Extensions Release 2. 1979. 5740-XE1.  
Program Product.  
Hardware performance improvements.  
Additional OS functions placed in microcode.
- MVS/SP1 - MVS System Product Version 1. June, 1980.  
Program Product.  
AKA MVS/SP1.  
Highest release: 1.3.6 or 1.3.7  
3375 and 3380 DASD support.  
Data Streaming channels for higher-speed DASD (above).  
Dual address space support (Cross Memory Services).  
Later renamed to "MVS/SP 1.1".
- MVS/SP2 - MVS System Product Version 2. June, 1980.  
Program Product.  
AKA MVS/SP2.  
GRS (Global Resource Serialization).  
CONSOLE function moved to its own address space.  
Later renamed to "MVS/SP 1.2".
- MVS/SP3 - MVS System Product Version 3. November, 1980.  
Program Product.  
AKA MVS/SP3.  
3033 (CPU) Extensions support.  
3081 CPU support.  
Channel queuing (I/O requests queued by the channel).  
I/O Suspend and Resume.  
Extended Swapping.  
Virtual Fetch for IMS (similar to VIO).  
Later renamed to "MVS/SP 1.3".

At some point in MVS/370, Extended Addressing was added, so that even though Virtual Storage addressing was still limited to 16 megabytes, the system could

use 32MB of real memory. This was later expanded to to 64MB. This made more real storage available to back up the virtual storage - all before MVS/XA. However, it was MVS/XA which provided "Virtual Storage Constraint Relief", being able to address 2GB per Address Space.

MVS, MVS/SE, MVS/SP ---> MVS/370  
 Renamed when MVS/XA was announced.  
 Non-XA MVS.  
 24-bit addressing (16 megabytes).

|  
|  
|  
|

V End of MVS/370 versions. The following versions are XA, ESA, etc.

MVS/XA - eXtended Architecture. 1983.  
 Aka "MVS/SP2" which followed MVS/SP 1.3.x.  
 Program Product.  
 Increased the addressing capabilities significantly.  
 31-bit addressing (2 gigabytes of address space).  
 Bi-modal addressing included (24-bit and 31-bit addressing).  
 DPS (Dynamic Path Selection for channels - up to 8 chans/dev)  
 DPR (Dynamic Path Reconnect for channels - up to 8 chans/dev)  
 Support for cached DASD controllers.  
 Improved Auxiliary Storage Manager (ASM).  
 Support for 3090-200 and 3090-400 CPUs.  
 Vector Facility.  
 Improved Task Dispatcher.  
 DIV (Data In Virtual) - Linear VSAM datasets.  
 Last release of MVS/XA: MVS/SP 2.2.3 (1989).

MVS/ESA - Enterprise Systems Architecture.  
 Aka MVS/SP3, MVS/SP4, MVS/SP5.  
 Program Product.

----- Versions of MVS/ESA -----  
 MVS/SP3... Highest release: Approx 3.1.3  
 Data Spaces (data in separate Address Spaces).  
 Hiperspaces (High-performance Data Spaces).  
 Hiperbatch.  
 DFSMS (Storage Management System for DASD).

MVS/SP4. 1990.  
 Sysplex  
 ESCON channels (Enterprise System Connection Channels - optical).  
 1990: 10 million bytes/sec transfer rate.  
 1991: 17 million bytes/sec transfer rate.  
 XCF.  
 Shared ESCON channels between LPARS.  
 HCD (Hardware Configuration Definition).  
 APPC/MVS.  
 DRM (Dynamic Reconfiguration Management).  
 PR/SM Automatic Reconfiguration.  
 Console integration.  
 SRM enhancements.

4.1: JCL enhancements -  
 IF, INCLUDE, etc added.

4.1.0 - unstable.  
 4.2.0 - unstable.  
 4.3.0 - good.  
 Highest release: 4.3.0

MVS/SP5 - Highest release of SP5: 5.2.2  
 Coupling Facility.  
 Parallel Sysplex - a single system image with  
 multiple systems, and easier expansion,  
 using the Coupling Facility.  
 SRM: "Goal Mode" for the Workload Manager.

OS/390 - MVS/SP6. 1996.  
 Program Product.  
 Re-integration of many separate products into 1 OS.  
 Price reductions for system software.  
 2 releases per year.  
 Easier Sysgens.

OS/390 3x - MVS/SP 6.x.x  
 Program Product.  
 Virtual Tape.  
 Smart Batch (parallel batch).

1999: FICON Channels - 100 million bytes/sec,  
 full duplex (read/write simultaneously  
 on same channel).  
 Can be as long as 10 kilometers (or more).  
 1 FICON channel can replace 8 ESCON channels.  
 (Newer FICON channels are even faster).

Recent release: OS/390 R10 = MVS/SP 6.1.0...

CVTPRODN = "SP6.1.0"  
 CVTPRODI = "HBB7703"  
 CVTVERID = " "  
 ECVTPVER = "021000"  
 ECVTPNAM = "OS/390 "  
 ECVTPOWN = "IBM CORP "  
 ECVTPIDN = "5647-A01"  
 ECVTPSEQ = X'00021000'

z/OS - MVS (OS/390) for Z-Series mainframes. 2001.  
 Program Product.  
 Uses 64-bit addressing.

## VM (VIRTUAL MACHINE)

### CP/40, CP/67, and VM (Virtual Machine)

- (CP/40) - Control Program for System/360 Model 40 using virtual storage. 1966.  
 --IBM internal product only. Not released to customers.--  
 Ran on a 360/40 with a DAT box running CP/40-CMS.  
 Developed by IBM at its Cambridge Scientific Center in Cambridge, Massachusetts.  
 CP: See below under CP/67.  
 CMS: See below under CP/67.  
 Ref: Bob Abeles.
- CP/67 - Control Program for the S/360 Model 67. 1967.  
 The 360/67 was a 360 Model 65 with a DAT box, giving it virtual-storage capability.  
 CP = Control Program. Also known as the "Hypervisor".  
 CMS (see below) enabled the "virtual machine" usability.  
 CP/67 with CMS was the first serious, large-scale virtual machine operating system.  
 It and VM were originally Open Source operating systems.
- CMS - Cambridge Monitor System.  
 - Conversational Monitor System.  
 Renamed by IBM from "Cambridge..." to "Conversational Monitor System".  
 This was the OS representing the user's virtual machine plus a time-sharing system, and ran under CP/67 (then VM).  
 Normally paired with VM as "VM/CMS".  
 CMS is the user's window into his virtual machine.
- VM/370 R1 thru R5 - Virtual Machine for System/370s.
- VM/370 R6 - Release 6.  
 \* Available via Bob Abeles and CBT at:  
 \* <ftp://ftp.cbttape.org/pub/cbttape/vm6/>
- VM/370 R6 BSEPP - Basic System Extension Program Product.  
 CMS full-screen and online HELP added.
- VM/370 R6 SEPP - Systems Extension Program Product.  
 Performance improved for guest OS's including MVS.
- VM/SP - System Product.
- SP ----> VM/Entry- VM/SP R3.1 - Entry level system.  
 Pre-genned. CMS only. 128 max CMS users.  
 4300 uni-processors. Object code only.
- SP ----> VM/IS - VM/SP R6.n . IS = Information Systems.  
 IBM Toronto. 1985/86.  
 The "Programmerless VM".  
 Dead-end product.
- VM/SP HPO - High Performance Option - early 1980s



- VM/ESA/370 - Non-XA version for 370 computers. 1990.
- Note: The VM/Assist Feature, if used, is for VM/370 series OS's - not VM/XA and above.  
VM/Assist is firmware microcode for portions of the VM/370 hypervisor.
- VM/XA MA - XA Migration Aid. Actually a version of VM/XA, not a utility. 31-bit addressing. For migration from MVS/SP1 (MVS/370) to MVS/SP2 (MVS/XA). Supports MVS/XA guests. SIE feature required. Low usage of product.
- VM/XA SF - eXtended Architecture / System Facility. 1986/87. Low usage of product.
- VM/XA SP - System Product. Approx 1987.
- VM/ESA - Enterprise Systems Architecture. 1990. When running in ESA mode, can run guest OSs which are running in 370 mode. Does not support 3330's or 3350's. Supports 3375's, 3380's, and 3390's in ESA mode. Requires SIE instruction [Start Interpretive Execution] to run. The "SIE I/O Assist" Feature is optional. When running in ESA mode, can run guest OSs which are running in 370 mode.
- z/VM - VM for zSeries mainframes. 64-bit addressing.

## TSS - TIME SHARING SYSTEM

### TSS - Time Sharing System

- TSS/360 - Time sharing system for Model 360/67 only. Uses virtual storage. A predecessor of TSO.
- TSS/370 - TSS for System/370 computers. Mentioned in the Bell Labs Technical Journal that they used TSS/370. See also "UNIX for System/370" in the non-IBM software section below.

## ACP AND TPF

### ACP and TPF - High Volume Transaction Processing

- ACP - Airline Control Program. Had a Hypervisor which allowed a guest job to run in V=R mode (Real mode), for times when ACP was waiting for work.
- TPF - Transaction Processing Facility. A further development of the Airline Control Program (ACP). Used by airlines, hotels, railroads, Visa, etc. Up to 7,700 transactions per second, according to one TPF'er. "Per second" is not a typo. Fast.

## MISCELLANEOUS OPERATING SYSTEMS

### OS's for the Model 360/44

---

OS/44 and PS/44 -

1960s.

OS's for the Model 360/44 scientific machine - a specialized System/360 for research and exploration.

Used mainly by academics and seismic exploration companies in the 1960s and 70s.

The 360/44 used register instructions and Floating Point instructions, and included special features such as data acquisition features, 32 pairs of extra PSW's, etc.

It had no decimal instructions, no BXLE, no SS formats, and some others, unless the Commercial Instruction Set Emulator was purchased. Ref: Rick Fochtman.

Speed of operation was enhanced by using fast memory access, optional high-speed circuitry, and hard-wired instructions rather than microcoded instructions.

Additionally, the precision of Floating Point operations could be reduced at the operator's control panel to provide faster speed for FP computations.

Had 32 extra Old/New PSW's in low-core.

The PS/44 OS was fairly primitive in terms of error recovery.

Could run a modified MFT, and run normal "commercial" programs with optional features installed.

Standard, high-level language: FORTRAN.

\* Not supported by Hercules. Non-standard machine.

### IX and AIX - IBM's Version of UNIX

---

IX/370 - Interactive Executive for 370s.

Ran as a guest under VM/SP on 4300s and up.

AIX/370 - Advanced Interactive Executive. First \*full\* attempt by IBM at running UNIX on the 370

AIX/ESA - Enterprise Systems Architecture - 1991

## Non-IBM Operating Systems

The operating systems listed in this section were written for use on IBM mainframe computers, but were not written by IBM.

### MUSIC - McGill University System for Interactive Computing

Approx 1972.

Developed from RAX (IBM operating system - 1966).

Purchased by IBM and renamed to MUSIC/SP.

Runs as a standalone OS or VM guest.

Very efficient.

MUSIC/SP - IBM's System Product version of MUSIC.

1985. IBM's name for the product:

"Multi-User System for Interactive Computing/System Product"

See <http://musicm.mcgill.ca/> for more info.

### MTS - Michigan Terminal System

An early time-sharing system initially developed in 1967 at the University of Michigan for use on the IBM S/360 Model 67. MTS was the first operating system to make production use of the virtual memory capability of the Model 67, and later to run on a dual processor version of the same computer. MTS was also used on the IBM S/370 series and compatible mainframes by a consortium of eight universities in the United States, Canada, and the United Kingdom over a period of 33 years (1967 to 1999).

See <http://www.clock.org/~jss/work/mts/index.html>

[http://en.wikipedia.org/wiki/Michigan\\_Terminal\\_System](http://en.wikipedia.org/wiki/Michigan_Terminal_System)

<http://archive.michigan-terminal-system.org/>

<http://www.clock.org/~jss/work/mts/index.html>

Hercules note: A version of MTS runs under Hercules today.

See: <http://archive.michigan-terminal-system.org/>

### UNIX for System/370 by Bell Labs

Bell Labs. 1979/1980.

Ran under a modified TSS/370 using a 3033AP, a 3033UP, a 3081K, and a 4341.

Used for 5ESS switch development using the 3033AP in 1981.

See AT&T Bell Laboratories Technical Journal,

Vol 63, Nbr 8, October 1984.

### TELPAR by Mike Green of SMU, and Dr. Robert McClure.

Late 1960s.

Used for EDA (Electronic Design Automation) and ATG (Automatic Test Generation) for electronics use and development. Written in a language called PL360. Enhancements made in the mid-1970s

including Sys/370 operation. Built-in spooling including spooling to memory. No IOGEN: Operator commands to declare new devices or new config. Stanford University has additional information.

Ref: Ron Tatum, Mark Waterbury.

**ORVYL** from Stanford University (<http://en.wikipedia.org/wiki/ORVYL>)

**VP/CSS** from National CSS (<http://en.wikipedia.org/wiki/VP/CSS>)

**UTS** from Amdahl ([http://en.wikipedia.org/wiki/UTS \(Mainframe UNIX\)](http://en.wikipedia.org/wiki/UTS_(Mainframe_UNIX)))

**PICK/370** from SMI and Ultimate - [http://en.wikipedia.org/wiki/Pick operating system](http://en.wikipedia.org/wiki/Pick_operating_system)  
Ran on S/370, 43xx, and 9370 machines.  
Interpreted code. 48-bit architecture.  
Ref: Peter Schaeffer.

---

## COMPONENTS AND FEATURES

### Glossary of Some Components and Features for DOS, OS/360, MVS, etc.

Short list of components and features and when first introduced for OS/360, its variants and progressions.

Many of these features are still used by MVS, OS/390, etc.

#### Some Basic Hardware and Addressing Modes:

Channels	- All operating systems	1965.
24-bit addressing	- 16 MB total per computer - OS/360	1965.
24-bit addressing	- 16 MB per address space - MVS	1974.
31-bit addressing	- 2 GB per address space - MVS/XA	1983.
32-bit addressing	- S/360 Model 67 only - CP/67	1967.
64-bit addressing	- 17 TB - z/OS	2001.

VSE - 64-bit addressing for OS only - z/VSE 4.3

VSE - 64-bit addressing for OS and apps - z/VSE 5.1

64 MB of main memory for systems running MVS/370,  
to provide additional real memory to backup  
the virtual memory, but with address spaces  
still restricted to 16 MB - MVS/370 197x.

Early Virtual Mode addressing - CP/67 1967.  
VS1 1972.  
VS2 (MVS) 1974.

DAT (Dynamic Address Translation) - 1967.  
Computer hardware for virtual storage.  
Often called a "DAT box".  
Translates virtual addresses  
into real addresses automatically.  
Used by all IBM "Virtual" OS's,  
not OS/360.

LPARs (Logical Partitions) - 1988.  
Partitions of memory, CPU, devices,  
etc, to give separate system images  
to groups of users, developers, etc,  
with each partition receiving a  
pre-set or variable proportion of  
the system's resources.  
For example, TEST, PROD, TECH partitions.  
Similar to VM, but done mostly in  
hardware. Formerly known as PR/SM.

.....

DASD (Direct Access Storage Device) -  
A disk drive.

RO/RI (Roll Out / Roll In) -  
First introduced: OS/MVT.  
Superseded by paging and swapping in VS.  
No longer available or needed.

## SMF (System Management Facility) -

First introduced: OS/360.  
 Free component.  
 Provides data on jobs, tasks, users,  
 datasets, tape mounts, IPLs, etc.  
 SYS1.MANX, SYS1.MANY..  
 Often used for billing purposes,  
 and general system information.

## ISMF (Interactive Storage Management Facility) -

First introduced: MVS.  
 Program Product.  
 Online, interactive interface to SMS.  
 For managing disk storage.  
 Full name: DF/SMS ISMF.  
 Part of SMS (not SMF).

## MF/1 (Measurement Facility/1) -

First introduced: Early VS2.  
 Free component of VS2.  
 Predecessor of RMF.  
 Used for reporting on system activity  
 at regular intervals.  
 Reported activity on the CPU, channels,  
 devices, performance groups, etc.

## RMF (Resource Measurement Facility) - 5740-XY4

First introduced: MVS.  
 Derivative of MF/1.  
 Program Product.  
 Optional product, but most shops have it.  
 Used for controlling, displaying and  
 reporting on system activity.  
 3 parts: RMF-1, RMF-2, RMF-3.

## HASP (Houston Automatic Spooling Priority) -

(Houston Automatic Spooling Program) -

First introduced: OS/360.  
 Optional. IBM wrote it for NASA. Approximately 1967.  
 Effectively replaced many of the functions of OS Readers,  
 Writers and Initiators, although those components still  
 existed.

Console commands prefixed with "\$" (or some other character).  
 Provided a backspace character for use with the console,  
 which the console hardware and OS lacked.  
 IE, there was no backspace key on the console keyboard, and  
 no character to indicate backspace, until HASP.  
 This was important for correcting mis-typed commands -  
 HASP and OS commands (a non-trivial issue).

HASP also added useful printer commands, such as backspacing  
 a printer a few pages (or more) for reprinting some pages  
 which had jammed in the printer. Printer setup operation  
 was also improved under HASP, for example, for printing  
 special forms, to get it right the first time.

Card readers and printers ran much faster under HASP.  
 For printing, unprintable characters were translated to blanks  
 by HASP, eliminating a wasted revolution of the print chain  
 looking for the non-existent character, and making the printer  
 run much faster for those situations.

HASP took advantage of "command chaining", printing multiple lines with one "Start I/O", with the CCW commands for each line chained together. This also freed up the CPU to work on other tasks, instead of repeatedly issuing a "Start I/O" for each line of printing.

(NOTE: It was odd that it took HASP, FDR (Fast Dump Restore), and Syncsort to utilize the "command chaining" and "Read Multiple Tracks" features of the computer hardware, before mainstream IBM software did - all either non-IBM or non-mainstream products, to the best of my knowledge. While IBM MIGHT have used these features somewhere, it was the utilization of advanced features of the 360/370 hardware that, in part, gave other products such better performance - and sales and usage.)

HASP Spool space for Readers and Writers was a single, permanent space instead of temporary spool files allocated, used, and deleted, allocated, used, and deleted, etc, greatly reducing the overhead of the system for Spooling operations.

HASP spooling also compressed the blanks (and possibly some other characters) on the Spool file, resulting in a much smaller Spool file than the plain OS version.

HASP used its own Automatic Spooling Priority system to prioritize the printing of spooled output, easing the workload on the operator - especially for managing the printing of reports which had a "due date and time" for completion.

Jobs were submitted by HASP to the OS Initiator.  
Also had an RJE facility (Remote Job Entry)  
and an NJE facility (Network Job Entry).  
Later renamed "Houston Automatic Spooling Program".  
Evolved into JES2.

OS Readers: Still needed for reading jobs on tape.  
OS Writers: Still needed for writing SYSOUT to tape.  
OS Initiators: Still needed for initiating jobs.

ASP (Asymmetric Multiprocessing System) -  
 (Attached Support Processing) -

First introduced: OS/MVT.  
 Optional. Not for MFT.  
 Derivative of DCS (Direct Coupled Systems) for  
 IBM 7090/7040 computers (1962...), prior to S/360.  
 Written by the same IBM group which wrote DCS  
 (IBM Los Angeles, Westwood).  
 Precursor of JES3.  
 Effectively replaced many of the functions  
 of OS Readers, Writers and Initiators,  
 although those components still existed.  
 OS Readers: Still needed for reading jobs on tape.  
 OS Writers: Still needed for writing SYSOUT to tape.  
 OS Initiators: Still needed for initiating jobs.  
 Jobs were submitted by ASP to the OS Initiator.  
 Created a loosely-coupled, single-system  
 image via channel-to-channel communication  
 between computers, and a shared Spool.  
 Also had RJP facility (Remote Job Processing).  
 With RJP, jobs could be routed to other  
 remote systems for execution, and the SYSOUT could  
 be returned to the local system for printing.  
 Main processor: The Global processor.  
 Other local processors: Local processors.  
 Name changed to "Attached Support Processing"  
 at some point.  
 Evolved into JES3.

JES (Job Entry Subsystem) -

First introduced: VS1 and VS2 R1 (SVS).  
 Readers, Writers, Initiators.  
 Referred to as "JES1" for VS1 when MVS was  
 introduced.  
 Integrated into the system.  
 No "\$" commands or messages like HASP or JES2.  
 JES for VS1 was later called "JES1" to  
 differentiate it from "JES2" and "JES3" which  
 became available for OS/VS2 R2 (MVS).  
 JES was always part of OS/360, but IBM  
 never publicly called it "JES" until VS1  
 was announced, although they called it  
 "JES" internally, and used the JES term with  
 some customers. Same story for "RES".

JES1 (Job Entry Subsystem 1) -

Rename of JES for VS1.  
 Same as "JES", but applied to VS1 only.

RES (Remote Entry Subsystem) -

First introduced: VS1.  
 An RJE subsystem for VS1 only.  
 "RJE" means "Remote Job Entry".  
 Used RTAM (Remote Terminal Access Method).  
 Superseded "RJE" for OS/360.



- JES2 (Job Entry Subsystem 2) -  
 (Choose JES2 or JES3 at MVS SYSGEN).  
 First introduced: MVS.  
 An enhanced version of HASP.  
 Uses "\$" commands and messages.  
 Communicates with the system via the subsystem interface (SSI). Linked separately.  
 The OS Reader/Interpreter became the JES2 Converter/Interpreter (or JES3 equivalent for JES3).  
 The OS Writer became the External Writer for writing SYSOUT to tape (non-spooled SYSOUT).
- JES3 (Job Entry Subsystem 3) -  
 (Choose JES2 or JES3 at MVS SYSGEN).  
 First introduced: MVS.  
 An enhanced version of ASP (not HASP).  
 Communicates with the system via the subsystem interface (SSI).  
 See ASP for more info.
- GAM (Graphics Access Method) -  
 First introduced: OS/360.  
 For graphics terminals (2260s, etc).  
 OS/360 only - not DOS.  
 DOS used BTAM for the same functions.
- RTAM (Remote Terminal Access Method) -  
 First introduced: VS1.  
 For VS1 only.  
 For some types of remote terminals.
- STRAM (Synchronous Transmission Access Method) -  
 First introduced: OS/360.  
 Also available for DOS.  
 For some types of remote terminals.  
 Used synchronous transmission - not Bi-synch.
- BTAM (Basic Telecommunications Access Method) -  
 First introduced: OS/360.  
 Also available for DOS.  
 Known as "Baker TAM".  
 Required by MVS 3.8J ..  
 BTAM/SP: BTAM / System Product.  
 First introduced: MVS/XA.  
 Program Product version of BTAM.
- QTAM (Queued Telecommunications Access Method) -  
 First introduced: OS/360.  
 Input and output messages were queued on disk.  
 Big and internally complicated for its time.  
 Not too efficient. Early teleprocessing.  
 Replaced by TCAM.

TCAM (TeleCommunications Access Method) -

First introduced: OS/360.  
 Replaced QTAM.  
 Used by TSO in MVT and MVS.  
 Required by MVS 3.8J ..  
 ACF/TCAM: Program Product version of TCAM.  
 First introduced: MVS/XA.

VTAM (Virtual Telecommunications Access Method) -

First introduced: VS1, VS2.  
 Superseded TCAM, although TCAM remained available for years and ran concurrently with VTAM for years.  
 Used by TSO in MVS.  
 Used more CPU time and memory than TCAM.  
 In VS1, used approx 900K for buffers and control blocks, and cycled thru its memory frequently so that the 900K was in effect "real memory", not virtual memory.  
 Early versions didn't include Pacing, but this was added later.  
 Talks to the Network Control Program (NCP) in a 370x controller, as well as to local controllers (3274, etc).  
 Known as "Victor TAM".  
 ACF/VTAM: Program Product version of VTAM.  
 First introduced: MVS/XA.

TSO (Time Sharing Option) -

First introduced: OS/MVT, 1971.  
 Never released for MFT, VS1, or DOS.  
 Line mode editor + utilities.  
 Predecessor was TSS (TSS/360, TSS/370).  
 Primitive interactive and editing capabilities compared with a full screen editor such as SPF, but had all the necessary functions.  
 SPF invokes TSO functions.  
 Some batch jobs also invoke TSO functions.

TSO/E (TSO Extensions) -

First introduced: MVS/SP1 (MVS/370).  
 Program Product.  
 Later version of TSO.  
 Adds TSO Command Package, TSO Session Mgr, TRANSMIT/RECEIVE, full-screen 3270 logon, improvements to HELP/ALLOCATE/CLIST, etc.  
 Later, PRINTDS and some RACF commands were added.

SPF (Structured Programming Facility) -

First introduced: MVS before 1978.  
 Was product number 5787-XT2.  
 Program Product.  
 Not for OS/VS1.  
 Full-screen editor + utilities. Used TSO.  
 Not related to "Structured Programming".  
 Meaningless name, but a great product!!  
 Rls 2.2: 1978. OUTLIST added. Popular version.  
 Later: New names and acronyms (ISPF, ISPF/PDF).  
 Now called: Interactive System Productivity  
 Facility / Program Development Facility.  
 User of TSO/E (enhanced TSO).  
 Available in PC versions and for other OS's  
 including VM, VSE, and non-IBM OS's;  
 These are functionally equivalent to SPF, but  
 other versions don't use TSO or TSO/E.  
 See: <http://www.planetmvs.com/spfeditor/>

ISPF (Interactive System Productivity Facility) -

First introduced: MVS.  
 See SPF. Often still referred to as "SPF".

BSAM (Basic Sequential Access Method) -

First introduced: OS/360.  
 For accessing records sequentially from  
 any sequential file, such as cards, tape, disk,  
 using Assembler. Uses READ/WRITE macros.  
 Requires more work from the programmer than  
 for QSAM.

QSAM (Queued Sequential Access Method) -

First introduced: OS/360.  
 For accessing records sequentially from  
 any sequential file, such as cards, tape, disk.  
 Uses GET/PUT macros in Assembler.  
 Automatic record deblocking and read-ahead.  
 Requires less work from the programmer  
 compared to BSAM.

BDAM (Basic Direct Access Method) -

First introduced: OS/360.  
 For accessing records by actual location (CHR)  
 or relative location (TTR).  
 CHR means "Cylinder Head Record".  
 TTR means (relative) "Track Track Record".  
 Difficult to use but very efficient.  
 Files created using CHR addresses are  
 "unmovable" due to their actual disk addresses,  
 and are so flagged on a directory listing.

**BPAM (Basic Partitioned Access Method) -**

First introduced: OS/360.  
DOS equivalent: Library structure.  
For accessing "libraries".  
A PDS is a dataset containing members (files).  
A dataset containing the files is the equivalent  
of a "Folder" containing files, in the PC world.  
SYS1.PARMLIB is an example of a PDS.  
There is no "QPAM" access method - that is,  
a PDS cannot be read sequentially, member by  
member, by specifying the DSN in a JCL  
statement. To read each member in the PDS,  
each member must be read individually via  
programming, using BPAM.  
Exception: IBM utilities can read an entire  
PDS.

**ISAM (Indexed Sequential Access Method) -**

First introduced: OS/360.  
Files used indexes, and records used keys  
for locating the appropriate record.  
The DASD hardware formatting of "Count, Key, Data"  
(CKD) supplied the structure for the use of  
hardware keys (on disk).  
Could be fast or slow depending on the  
application and file design.  
Preformatting files with dummy records (if possible)  
avoided the slowness of inserting new records,  
making it an excellent access method.  
Still supported by OS/390, but rarely used.  
VSAM supports the same functionality.

**VSAM (Virtual Storage Access Method) -**

First introduced: VS1, VS2. (Also DOS/VS).  
Intended as a replacement for ISAM, BDAM,  
BSAM and QSAM for disk files.  
Sequential, indexed and relative files  
(ESDS, KSDS, RRDS).  
Runs only on VS systems (not OS/360 for  
example).  
Utilities: IDCAMS (Access Method Services).  
Is not related to "virtual storage": Simply  
introduced with the virtual operating  
systems, and they had to call it "something".  
Features: Supports indexed files better than  
ISAM, sequential files worse than QSAM,  
direct files worse than BDAM, according to  
one Herculean's opinion.

## OS Catalogs (CVOL Catalogs) -

First introduced: OS/360.  
 The original "dataset" catalogs for OS/360, cataloging disk and tape files, including Generation Dataset Groups (GDGs).  
 Non-VSAM. Easy to use. Good performance.  
 Faster than VSAM catalogs and ICF catalogs.  
 A Catalog is on a CVOL (Control Volume).  
 The Master Catalog is named SYSCTLG.  
 Each User Catalog connected to the Master Catalog is also named SYSCTLG (1 per volume).  
 Called "CVOL Catalogs" later, when VSAM catalogs were introduced in 1972.  
 Used JCL or IEHPROGM to manipulate.  
 Same basic concepts still apply today.  
 No dates in the catalog, but unusable by MVS/ESA and OS/390 after 12-31-1999 due to certain changes made in MVS/ESA and OS/390. (VTOCs contain dates, VSAM and ICF Catalogs contain dates, but OS Catalogs don't contain any dates. Y2K was not an issue for OS Catalogs. Disabled by IBM for use after 12-31-1999.)

## VSAM Catalogs - First introduced: VS1, VS2.

Intended as a replacement for OS Catalogs although OS Catalogs hung around for years.  
 KSDS format (keyed). "Owns" volumes.  
 Needed for cataloging VSAM clusters, with additional VSAM info such as key size, key location, Control Interval size, statistics, etc.  
 Advantage over OS Catalogs: Compatible with DOS/VS, DOS/VSE, VS1, VS2, MVS (and above).  
 Disadvantage: Much slower than OS Catalogs.  
 Not considered Y2K compliant for MVS/ESA and OS/390 by IBM, and inoperable after 12-31-1999.  
 Replaced by ICF Catalogs.

## ICF Catalogs (Integrated Catalog Facility - VSAM) -

First introduced: MVS.  
 Uses VSAM. KSDS and ESDS structure.  
 The ICF is:  
   The Basic Catalog Structure (BCS).  
   The VSAM Volume Dataset (VVDS).  
 The BCS is a VSAM KSDS file (keyed).  
 The VVDS is a VSAM ESDS file (sequential).  
 Replaced previous version of VSAM catalogs.  
 Component: Originally DF/EF.  
 Does not "own" volumes.  
 No dataspace. All VSAM objects are UNIQUE.  
 Multiple catalogs per volume OK.  
 Much faster than plain "VSAM Catalogs".  
 ICF is required for MVS/ESA and OS/390 as of 01-01-2000.

- Page files - First introduced: VS1 and VS2.  
 For virtual storage OSs: VS1, VS2 and above  
 (and DOS/VS and above).  
 For MVS and above: Uses VSAM structure but  
 not VSAM I/O (uses its own Paging I/O method).  
 MVS: Files defined as PAGESPACE via IDCAMS.  
 VM: Uses a "page area".
- Swap files - First introduced: MVS.  
 Similar to Page files and used in addition to  
 them.  
 Used for swapping entire Regions into or  
 out of memory in 1 "swap" operation.  
 For MVS and above - not VS1 or DOS/VS.  
 Uses VSAM file structure but not VSAM I/O  
 (uses its own Paging access method).  
 MVS: Files defined as PAGESPACE via IDCAMS.
- SDSF (Syslog Display and Search Facility) -  
 (Spool Display and Search Facility) -  
 First introduced: MVS.  
 Program Product.  
 Displays system activity, SYSOUT listings,  
 console log, can use as a console, etc.  
 More powerful than product name suggests.  
 Excellent product!  
 Originally a Field Developed Program.  
 Later renamed "Spool Display and Search Facility".
- SDF (Screen Definition Facility)  
 A screen mapping tool used by a programmer when  
 developing a CICS program.
- OSMP (OS Maintenance Program) -  
 First introduced: OS/360.  
 For applying system maintenance.  
 Predecessor of SMP.  
 Had very basic functions compared to SMP.  
 PGM=IHGUAP (called the "Update Analysis"  
 Program").  
 Program number 360S-UT-506.  
 Pub: GC27-6918-3, 1970, is for OS Release 18.
- SMP (System Modification Program) -  
 (System Maintenance Program) -  
 First introduced: OS/360 (HMASMP).  
 For applying system maintenance and  
 installing new OS releases or program  
 products.  
 RECEIVE/APPLY/ACCEPT processing.  
 VS: Used significant amounts of memory below  
 the 16 meg line, sometimes causing problems  
 in being able to run it at all if memory  
 below the line was tight.  
 APPLY: Always use "DIS(WRITE)" for speed.  
 SMP 4.24: New syntax introduced.  
 (DOS/VSE version of "SMP": MSHP).  
 (VM/SP version of "SMP": SES).  
 At some point, the name was changed to  
 "System Maintenance Program."  
 Pub: GC28-6791-2, 1976, is for OS Release 21.  
 MVS 3.8J came with SMP 4.45..

## SMP/E (SMP Extensions) -

First introduced: MVS.

Improved version of SMP.

VSAM files. No CDS or ACDS, per se.

Does 1 linkedit per dataset (such as  
SYS1.LINKLIB) rather than 1 linkedit per  
module, if possible.

Bundles related updates and commits them.

No memory problems with trying to obtain  
large amounts of memory below the 16 MB line.

## MAINFRAME/PC COMPARISONS

### For PC people NEW to mainframes

For those of you who are NEW to the world of IBM mainframes, and are familiar only with PCs, please be aware of a few major differences between PCs and IBM mainframes (there may well be dozens of other major differences I'm not thinking of):

#### 1. Data encoding scheme.

In general:

Mainframes use EBCDIC (8 bits) - hex 00 thru hex FF.

PC's use extended ASCII (8 bits) - hex 00 thru hex FF.

Extended ASCII is standard ASCII (7 bits) plus 1 more bit.

EBCDIC: blank=40, A-I=C1-C9, J-R=D1-D9, S-Z=E2-E9, 0-9=F0-F9.

Numbers collate higher than letters.

Low to high: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

			0 F0	blank 40
A C1	J D1		1 F1	. 4B
B C2	K D2	S E2	2 F2	, 6B
C C3	L D3	T E3	3 F3	/ 61
D C4	M D4	U E4	4 F4	\$ 5B
E C5	N D5	V E5	5 F5	& 50
F C6	O D6	W E6	6 F6	* 5C
G C7	P D7	X E7	7 F7	( 4D
H C8	Q D8	Y E8	8 F8	) 5D
I C9	R D9	Z E9	9 F9	= 7E

A System/360 reference card (green card) or System/370 reference card (yellow card) or System/390 reference card (white card) has the details.

#### 2. Transfer rates: Bits versus Bytes, and multiple devices.

PC transfer rates have historically appeared to be blazingly fast because the rates were often expressed in bits per second (bps) instead of bytes per second (BPS). This was just a marketing ploy to make PCs, modems, and LANs appear to be extremely fast.

You need to divide the bit rate by 8 to get the byte rate - the rate used by mainframes, in most cases. (Modem speeds are listed as bits/second).

Bytes/second (BPS) = bits/second divided by 8.

Bits/second (bps) = bytes/second times 8.

While recent PC devices are quite fast, the older ones were slow.

2400 bps = 300 bytes per second. Extremely slow!!

56,000 bps = 7000 bytes per second. Extremely slow!!

10,000,000 bps = 1.25 million bytes per second. Still slow!!

When you use the same measuring units, you see that some of the older mainframe peripherals were quite fast. And MULTIPLE devices were running at the same time!!



## Example: The New Company LAN versus an Old Mainframe Tape Drive:

PATHETIC	MUCH FASTER
New Company LAN, 1993.	Old 3480 Mainframe Tape Drive, 1985.
10 million BITS/sec.	24 million BITS/sec.
1.3 million BYTES/sec	3 million BYTES/sec.
	(Cartridge drive).

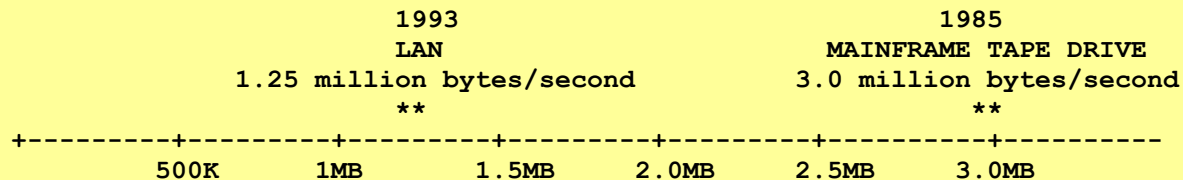
The corporate LAN group was convinced that their new LAN was blazingly fast (10 megabits/second) compared to the mainframe's I/O speeds. They were strutting around the office, bragging about their new toy which they thought had trounced the mainframe in terms of speed.

Ignorance is bliss.

They didn't have a clue what they were talking about. 20-year old mainframe devices from the 1970s were just as fast, while current and recent mainframe devices were even FASTER.

Additionally, many of the mainframe devices could operate concurrently using multiple channels, compounding the total thrupt of the mainframe. The mainframe ate the LAN's lunch - even with old hardware.

Always remember Rule Number 8: Divide bit rates by 8 to get byte rates, and Rule 48, multiply the mainframe I/O device rates by the number of possible concurrent device operations (I/O's).



Device Transfer Rates in Bytes Per Second.  
New LAN vs. Old tape drive

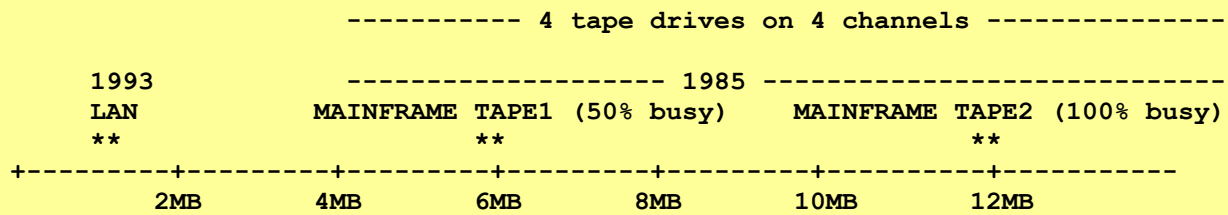
MAINFRAME TAPE DRIVE = 3480 Mainframe Cartridge Tape Drive from 1985

The LAN's capacity was 10 million bits per second (1.25MB/sec).

A single 3480 mainframe tape drive, made in 1985, was more than twice as fast as the LAN.

We had around 16 such tape drives at that company.

With 4 channels attached, 4 tape drives could be reading or writing simultaneously, and the tape thrupt would have been about 12 million bytes per second at 100% device-busy and channel-busy, and 6 million bytes per second at 50% device-busy and channel-busy. Reasonable?? Yes. Typical?? Yes.



Device Transfer Rates in Bytes Per Second

LAN	(1.25 MB)	=	1 new LAN running at 100% busy.
TAPE1	(6.00 MB)	=	4 old 3480 tape drives running at 50% busy channel-busy.
TAPE2	(12.00 MB)	=	4 old 3480 tape drives running at 100% busy channel-busy.

These **\*\*OLDER\*\*** mainframe tape drives typically delivered 4-8 times more throughput than the **\*\*NEW\*\*** LAN.

## THE OLD TAPE DRIVES ATE THE LAN'S LUNCH!!

6 megabytes per second.

12 megabytes per second.

Versus a paltry 1.25 megabytes per second for the new LAN.

## NEWER TAPE DRIVES WERE EVEN FASTER!

## NEWER DISK DRIVES WERE EVEN FASTER THAN THAT!

## THEY ALL ATE THE LAN'S LUNCH, BREAKFAST, AND DINNER!!

Rule Number 1: Know what you're doing.

Rule Number 8: Divide bit rates by 8 to get byte rates.

Rule Number 48: Multiply the mainframe device byte-transfer-rates by the number of devices which can **CONCURRENTLY** perform I/O's, to obtain the maximum possible data rate figure.

Note: For Rule 48, expertise and an in-depth knowledge of the system configuration is required.

*What was realistic for tape drive and tape channel usage? At a different company, measuring channel-busy for the 2 tape channels on an IBM 3081-GX multi-processing mainframe running MVS with 13 tape drives in 1987, I found that during the Evening shift (which saw the heaviest use of tape drives and tape channels), using slower 3420-8 tape drives, the monthly average channel-busy was around 50%, with frequent peaks at 95% channel-busy, lasting 1 hour.*

*So, the numbers listed above are realistic for a faster CPU in the 1980s/1990s, a bigger workload with heavy tape usage, 16 tape drives, 4 tape channels, with a maximum of 4 tape drives simultaneously transferring data: 50%-100% channel busy.*

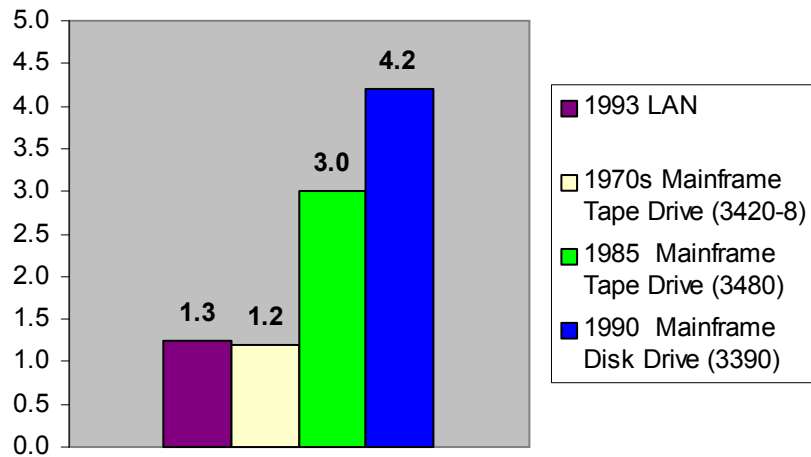
*(At a later datacenter, the mainframe computer had 48 channels, total).*

*Comparing apples-to-apples using the max sustainable device transfer rate as the benchmark, that rate would be close to 12MB/sec for the 4 3480s on 4 channels - 8 times faster than the max sustainable rate of the LAN.*

Using data from 1987 at the company using much older 3420-8 tape drives (transfer rate = 1.2MB/sec) and only 2 channels, at 100% channel-busy the transfer rate would have been only 2.4MB/sec - still twice as fast as the new 1993 LAN.

The transfer rates shown in the graphs are based on DEVICE transfer rates (actual transfer rates) - not the maximum speed of the channel.

**Older Mainframe Devices  
versus 1993 LAN Speed  
in Max Megabytes per Second  
per Device**



3420-8 Tape Drive (1970s)



Group of 3420-8 Tape Drives (1970s).

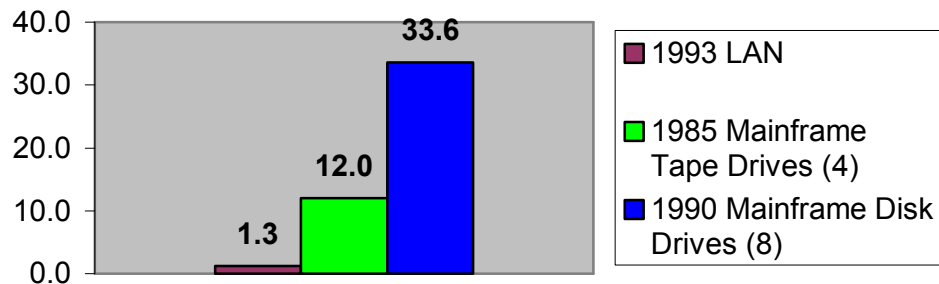


3480s (1985) and 3420s (1970s).

The 3480s (foreground) used cartridge tapes.

2 3480 tape drives (cartridge tape drives) are shown in front of the older 3420s (reel-to-reel tape drives).

**Using Older Mainframe Devices:  
CONCURRENT DEVICE THRUPUT  
versus 1993 LAN thrupt  
in Max Megabytes per Second -  
4 Tapes on 4 Channels at 100% Busy  
(green),  
8 Disks on 8 Channels at 100% Busy  
(blue)**



The mainframe thrupt rate was enormous compared to the 1993 LAN.  
The Comm folks only had 1 LAN.  
The mainframe had 16 tape drives, dozens of disk drives, printers,  
and 48 channels (paths) to the devices, etc.  
New LAN: 1.3 MB/sec. 4 old tape drives: 12.0 MB/sec.  
8 old disk drives: 33.6 MB/sec.  
**Tape+Disk: 45.6 MB/second.**  
**Message to the mainframe: OPEN BAG, EAT LAN'S LUNCH.**

### 3. Looping.

Mainframes don't loop. They wait.

Mainframes do not loop until an I/O operation (or some other operation) is finished, tying up the CPU at 100% busy. They either do other work for other people, Jobs, or tasks, or they WAIT, if there's no other work to do.

There is a hardware state called the "WAIT" state, invoked when there's no more work to do. When a device (or timer, etc) signals that it has completed its task (read a record, etc), the computer wakes up and handles it.

If a programmer codes a loop running under MVS, the Workload Manager will not permit that program to monopolize the CPU, and will give control to other tasks, as needed.

PCs are supposed to be multi-tasking. But some programs on my PC appear to loop until finished, preventing me from doing anything else. It won't let me multi-task. This is VERY user-unfriendly.

Mainframes are polite, and achieve huge throughput rates.

Exception: Some new IBM programmers have coded "spin loops" in the operating system, at various places. In at least one case - obtaining a CTC channel in a busy Sysplex - it caused so many problems tying up the

computer when the channels were busy, that it had to be changed to a WAIT - just as it should have been coded in the first place. I guess they slept through the "multi-tasking" class....

#### 4. Intelligent Hardware: Channels, Control Units, and Devices.

Channels are the main KEY to the mainframe's enormous data throughput.

For high throughput on a computer, many factors are important such as the speed of the CPU/CPUs, the number of CPUs, the amount of memory, the speed of the devices, the rate at which work can be submitted, etc. Channels are important because they allow multiple I/O operations to take place simultaneously, mostly without CPU involvement.

Channels are intelligent work horses, and are VERY IMPORTANT! They are hard-working mini-computers in the mainframe, and they offload most of the I/O processing from the CPU.

Control units connecting the devices to the channels also have built-in intelligence, as do the devices themselves. IBM mainframes use channels and control units for I/O with devices.

A channel is:

- A physical PATH to the device.
- A mini-computer which LOCATES the data.
- A minicomputer which MOVES the data.

The mainframe's CPU usually needs to do relatively little work to read or write a record. Once the channel program is started, the channel takes over the job of locating the record, and transferring the data.

("Locating the data" applies mainly to disk I/O operations).

Then the CPU waits for the I/O to complete, or performs other work for other programs while the I/O is taking place.

And for the past 25 years or more, channels have even selected the PATH to the device, so the OS wouldn't have to.

This is why tapes can be spinning, printers can be printing, yet the CPU is loafing, with the WAIT light glowing brightly.

A tape-to-print operation is trivial for a mainframe.

Copying "huge file X" to file Y is also trivial.

Sorting hundreds of thousands of records can be trivial.

A mainframe with 8 CPUs and 48 channels has 56 processors to do the work, plus smart electronics built into each control unit and device (although an IBM spec sheet will list just the 8 CPUs as Processors).

(The IBM PS/2 micro-channel worked well for PCs, but didn't last long due to its proprietary nature and steep licensing fees for its use.)

To start an I/O operation, the OS issues a STARTIO instruction, or a variant of it. The channel then loads the first CCW (Channel Command Word) into its memory and starts the operation, independently of the CPU. When the I/O has completed, the channel interrupts the CPU and presents the status of the operation to the OS.

For 3-digit device addresses, the format is x'cuu' where cuu is the entire device address, 'c' is the channel, and 'uu' is the unit.

Example: '152' is a device on channel 1, while '52' is the control unit and device. With multiple channel paths to a device (very common), the device might be known as "152/352" for example, meaning "connected to channels 1 and 3".

## 5. Printing.

The old mainframe printers were fast and **HEAVY DUTY!**

Even the oldest impact printers from the 1960s and 1970s were fast, and often ran almost continuously, except when changing the paper, lining up the forms, and replacing ink ribbons.

The line width varied, but was typically between 120 and 132 chars.

In the chart below, I assume the printer is running only 20 hours/day - not 24 - with 4 hours/day for changing paper, cleaning, etc.

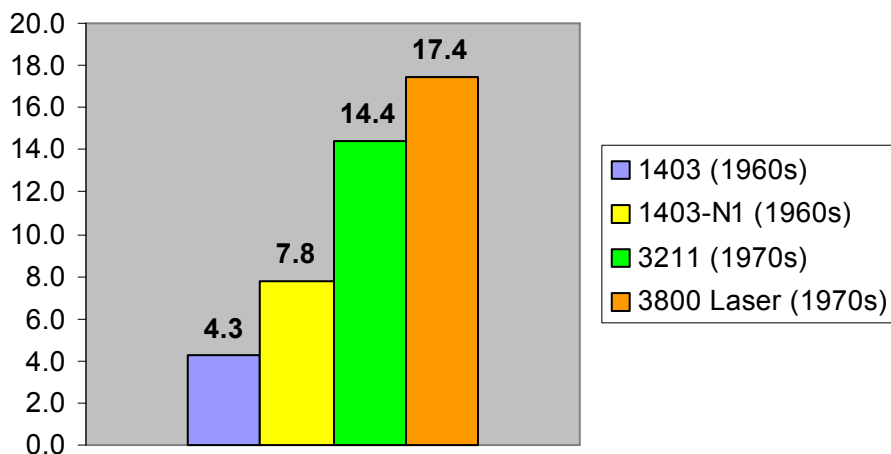
Printer	Type	Lines	Pages	Pages	Pages	Era
		Per-Min	Per-Min	Per-Week	Per-Year	
1403	Impact	600	10	84,000	4.3M	1960s
1403-N1	Impact	1100	18	151,000	7.8M	1960s
3211	Impact	2000	33	277,000	14.4M	1970s
3800	Laser	??	40+	336,000	17.4M	1970s

For a large company in the 1970s, with 4, 3211 impact printers, running 20 hours/day, the printed output could be 158,000 pages per day, or over 1 million pages per week, or 57.5 Million pages per year.

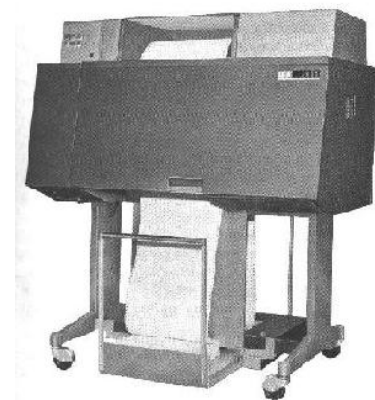
Even 2 of the old reliable 1403-N1 impact printers from the 1960s could have produced 302,000 pages per week, or over 15M pages per year. Not bad for a couple of old, mechanical devices.

Laser printers from the 1970s could have produced much more.

### Mainframe Printer Output in MILLIONS of Pages per Year Per Printer Running 20 hours/day



Even ONE old 1403 printer from the 1960s could produce 4.3 million PAGES of output per year. Large datacenters always had several printers. Note that the advanced, high-speed 3800 laser printer was in use in the 1970s - before the PC was invented.



1403 Printer (1960s)

What about the CPU load of 1, 2, 3, or 4 printers constantly printing at a fairly high rate of speed: Wouldn't that overload the CPU? Not at all. The CPU would need to retrieve the records from the spool file and print them - trivial work since large "blocks" of print records are read, and the print records would be in a compressed form and smallish, and would place hardly any burden on the CPU. The WAIT light on the mainframe would be glowing brightly if nothing else were running.

In fact, using HASP, JES2, and other spooler programs, the print commands for printing each line (sent to the channel, control unit, and printer) are chained together for printing an entire page, whenever possible, requiring just one "START I/O" instruction from the operating system per page - at least for the "line" printers.

Interrupts from the channel, control unit, and printer would also occur only once per page. The channels, control units, and printers would do most of the heavy lifting, while the CPU loafed.

#### 6. Buffer overflow.

There is no such thing as a "buffer overflow" on a mainframe.

"Buffer overflows" are a constant source of security problems for Microsoft Windows, and have caused Microsoft and its users VERY serious problems. Patches have been released for over 10 years for buffer overflow problems.

But there is no such thing as a buffer overflow on a mainframe, unless someone were coding at the READ/WRITE level (basic) and was dumb enough to specify a buffer smaller than his READ request.

Naturally, if you try to read a 100-byte record (or block) into an 80-byte buffer (I/O area), the last 20 bytes of the record will clobber whatever comes after the buffer area in your program. Most programmers don't program at the basic READ/WRITE level, are not that careless, and any such problems would quickly be discovered in testing. I've never seen anyone do it nor heard of such a thing in the mainframe world.



## 7. Overall Power.



Piper Cub



Boeing 747

*"Hello, Boeing? You can come over and pick up your 747 airliner. Someone decided to replace it with a Piper Cub."*

Since the introduction of the 386-based PCs, people have taken their PCs more seriously, and have done some serious work with them. That includes me!! Today's Pentium processors, large memories and disk drives are a quantum leap from what they were in the 1980s. However, many people have said since the late 1980's:

User in 1985: *"I have more power on my desktop than the mainframe at my company..."*. Me: *"Really? How would you know???"*

The mainframe in the computer room might be a large 3084-QX, or a 3090-600J, with 700 TSO users and 700 terminals attached to the mainframe, hundreds of online CICS users with hundreds more attached terminals, dozens of jobs running, tapes spinning (and cataloged and stored), multiple printers producing hardcopy 24 hours/day at high speed, etc.

But the PC user has a computer system on his desk containing only:

- A 386/SX25 CPU (25 megahertz clock rate - entry-level PC).
- 2 megabytes of memory.
- 1 diskette in the floppy drive.
- 1 hard drive (100 megabyte capacity - half empty).
- 0 tape drives (none).
- 1 user logged on (just him) doing a spreadsheet on the screen, waiting for input.
- 1 pokey, dot-matrix printer with continuous forms which has produced only 10 pages of output all day!!

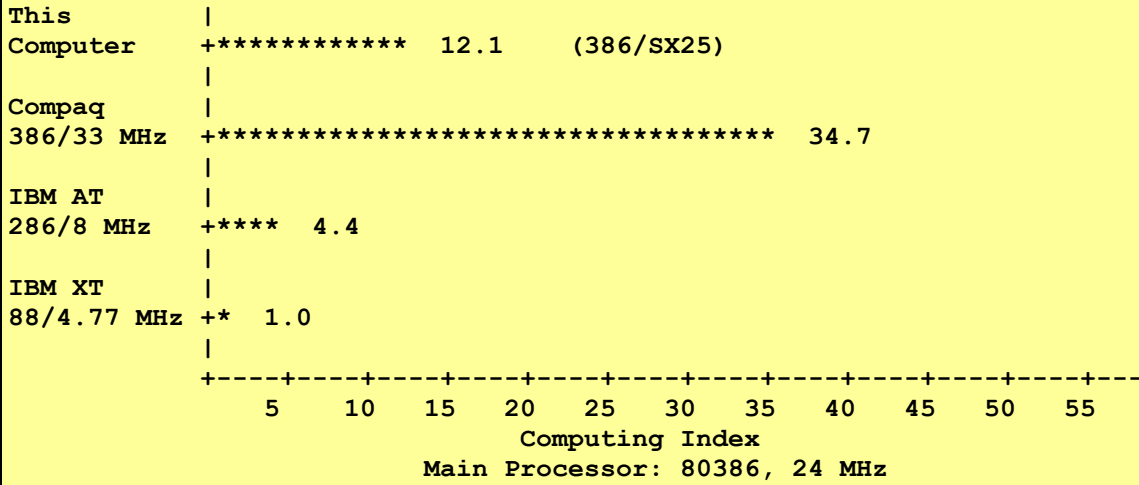
And when he DOES print something, his PC locks up like a vault until the printing has finished.

Why didn't management just run down to Computer City, buy a PC and some peripheral devices for \$2000-\$5000, plug it in, and unplug the mainframe, saving the company millions of dollars per year?? *"Hello, IBM? You can come over and pick up your mainframe and all the peripheral devices. We've decided to replace it with a Packard-Bell 386 PC and a little dot-matrix printer which overheats all the time."*

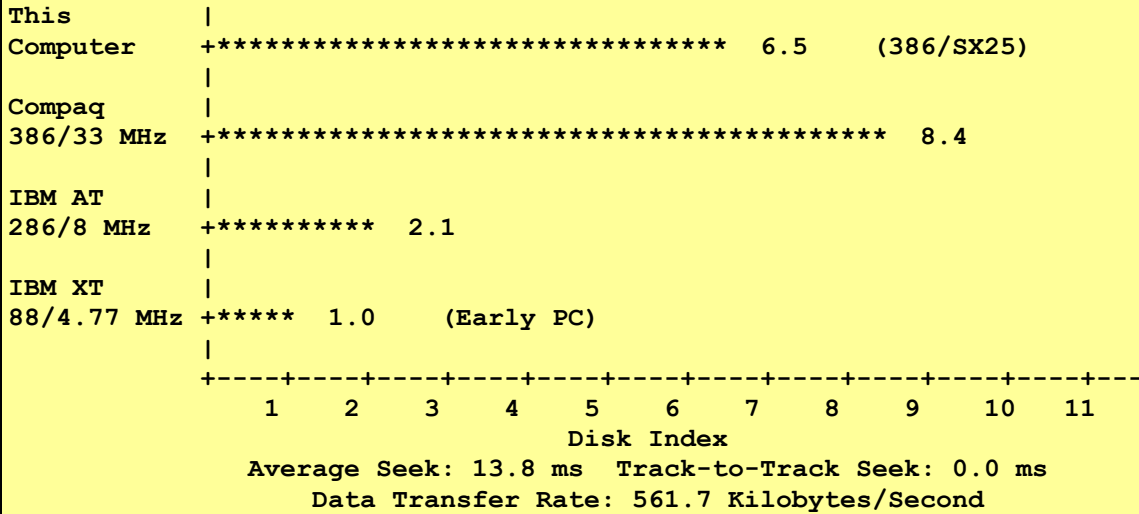
Obviously, they had excellent reasons for \*NOT\* doing that. As someone once said, "There's a reason mainframes are so expensive." It's the hardware, the software, and so much more... It's the total power!

Here are some graphs produced by Norton Utilities in 1992, measuring an early PC of mine (a Packard Bell 386/SX25) at a time when people were saying, *"I have more power on my desktop than the mainframe at my company..."*.

\*\*\*\*\*  
\* CPU Speed \*  
\*\*\*\*\*



\*\*\*\*\*  
\* Disk Speed \*  
\*\*\*\*\*



```

*****
* Overall Performance Index *
*****

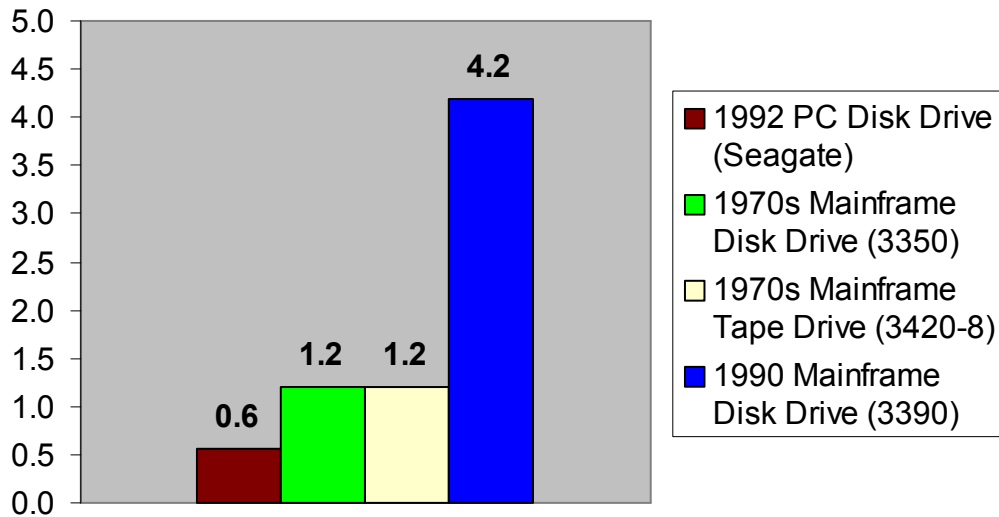
This      |
Computer  +***** 10.1   (386/SX25)
          |
Compaq    |
386/33 MHz +***** 25.9
          |
IBM AT    |
286/8 MHz +***** 3.7
          |
IBM XT    |
88/4.77 MHz +* 1.0   (Early PC)
          |
          +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
          5     10    15    20    25    30    35    40    45    50    55
          Overall Performance Index

```

Clearly, my first "PC home computer" wasn't as fast as some others such as Compaq computers, but it was a typical entry-level, affordable PC (the Compaq's were quite expensive, as I recall). But it was about 10 times as powerful as an early generation IBM PC - the IBM XT PC.

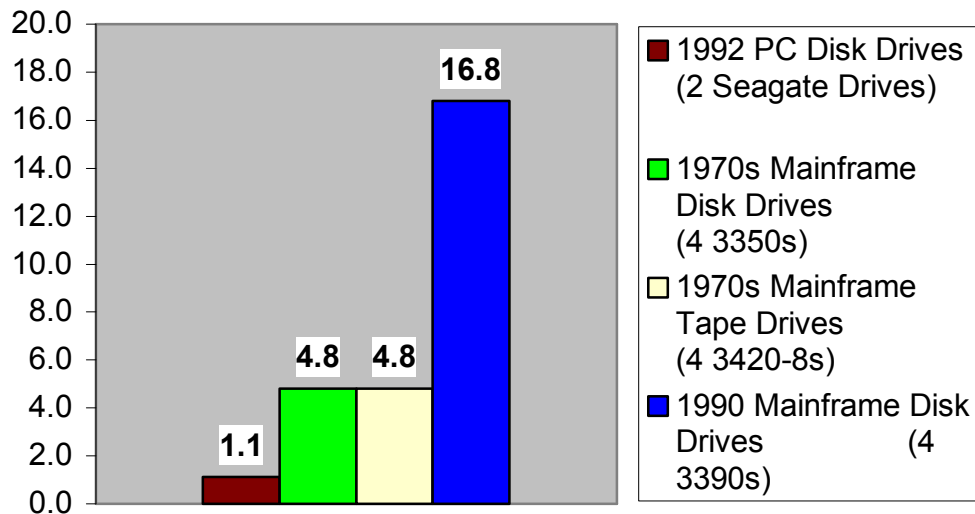
Assuming, for the moment, that my PC had been upgraded to 2 disk drives, and that they could transfer data to and from memory simultaneously by using the DMA hardware (unknown), running at 100 percent busy all the time, we see the following transfer rates for 1 and 2 PC disk drives, versus older mainframe disk drives and tape drives (using BENCHMARK thruput rates):

### PC and Mainframe Max I/O Rates (For Older Devices) in MB/Second



The older 1990 mainframe disk drive (3390) was 7 times as fast as the newer 1992 PC disk drive (Seagate)

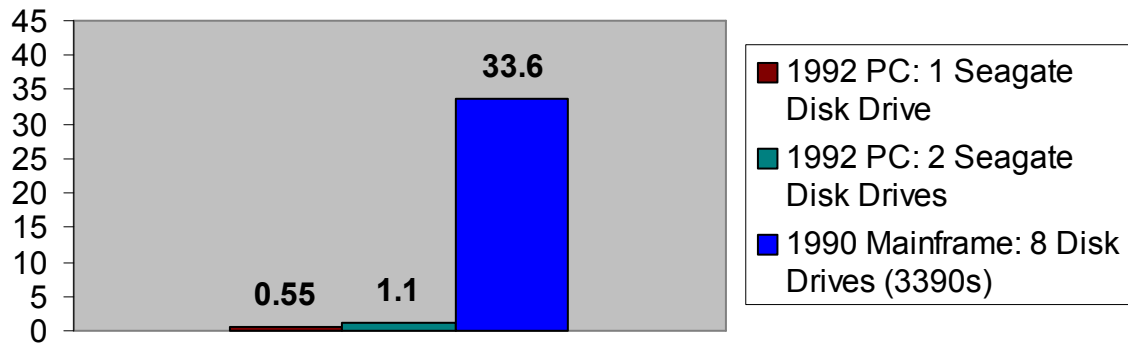
### Aggregate I/O Rates in MB/Second with 4 devices and 4 Channels at 100% Channel Busy



The mainframe tape and disk drives left the PC in the dust.

We assume that the PC disk drives are running at 100% busy, so we should assume the same for the mainframe channels and devices.

## Disk I/O Comparison: Aggregate PC/Mainframe Disk I/O Rates in MB/Second (with 8 mainframe channels at 100% Channel Busy)



With 8 mainframe disk drives active, and 8 channels, they would have transferred 30 times as much data as 2 PC disk drives, or 61 times as much data as 1 PC disk drive each second.

Just a sample of I/O rates - individual and aggregate - comparing a new PC disk drive from 1992 (running at about 100% busy) to old and new mainframe disk and tape drives should demolish the notion that "I have more power on my desktop than the mainframe at my company..." during that era.

Additionally, the highly efficient mainframe operating systems usually make mainframes far more nimble than their PC counterparts.

We'll skip the comparisons of CPU power (with multiple processors), memory, printing, disk capacity, tape capacity, access methods, databases, security, integrity, reliability, backup, number of concurrent batch jobs, number of online users, etc, etc.

Personal computers are great (I own several), and they're far more powerful today than they were in the 1980s or 1990s, but the "PC/mainframe power myth" which began in the 1980s is a complete falsehood. A comparison of a little Piper Cub airplane to a Boeing 747 comes to mind - or would you like to compare a Current PC to a Current mainframe Sysplex with 32 CPUs per mainframe, 240 fiber-optic (FICON) channels per mainframe, 200+ disk drives, etc... Mainframes have always been FAR more powerful than PCs (for the corresponding era), and still are.

(IE, a mainframe produced in 1981 could whip a PC produced in 1981, and a mainframe produced in 2009 could whip a PC produced in 2009, in terms of total power).

## 8. Features.

Many features considered to be "advanced" or "modern" on PCs are usually features found on mainframes of the 1960s-1980s under a different name, and still in use today. And many mainframe features are not found in PCs due to expense or the lack of need.

Examples:

IBM Mainframe	PC/Server (Windows and Intel oriented)
-----	-----
Online users - 1967	Client/Server - 1980s/1990s
Virtual storage - 1967	Windows with paging file (1988?)
Overlay or transient routine - 1965	DLL - Windows
Input/Output Channels - 1965	N/A (a DMA channel is close)
Memory Write Protection - 1965	Intel 486 - 1990s
Memory Read Protection - VS1-1971	Intel 486? - 1990s
Wait state - 1965	Intel Pentium - 1990s
Decimal Instruction Set - 1965	Intel 486 or Pentium - 1990s
Low Address Memory Protection	??
Supervisor Mode - 1965	??
Key-Zero Mode - 1965	??
Time-Slicing - 1965	Windows 95??
Multiple page and swap files - 1971	No
Performance Groups	No
Dual Address Spaces	No
Interpretive Execution	No
Service Processor	No
Program Event Recording	No
LPARs - 1988	No
Sysplex	No
Spooling (print and card)- 1966	Print spooling
Multiprogramming - 1965	Windows 95 - 1995
Multiprocessing-Tight - 1966 approx	199x
Multiprocessing-Loose - ASP-1965	No
System catalog - 1966	No (Windows Registry is close)
Generation Data Groups - GDGs-1966	No
Checkpoint/Restart - 1968 approx	Windows XP - Some capability (2002)
Standalone Utilities - 1965	No
SMF	No
RMF-1	No
RMF-2 - 1980?	Windows XP Task Manager - Partial capability (2002)
RMF-3	No
SDSF - 1978 approx	No
TSO - MVT - 1960s	No
SPF - 1977 approx	No (Partial with User products - 199x)
CICS	No
VSAM	No
IMS	No
RACF (security)	No
Authorized libraries (security)	No
VM	2000's
SMS (Storage Management)	No
Disk read-only switch	No
Laser Printers - model 3800 - 1970s	1990s
Color graphics terminals - 1970s	1980s
Disk RPS - 1970s	??
Channel Disconnect/Reconnect- 1970s	??
Disk caching - 1980s	1990s
Remote Job Entry - 1960s	Remote Procedure Call? - 2002

## 9a. Mainframe Replacement - Are These Really Mainframes???



360/20  
Mini



1130  
Mini

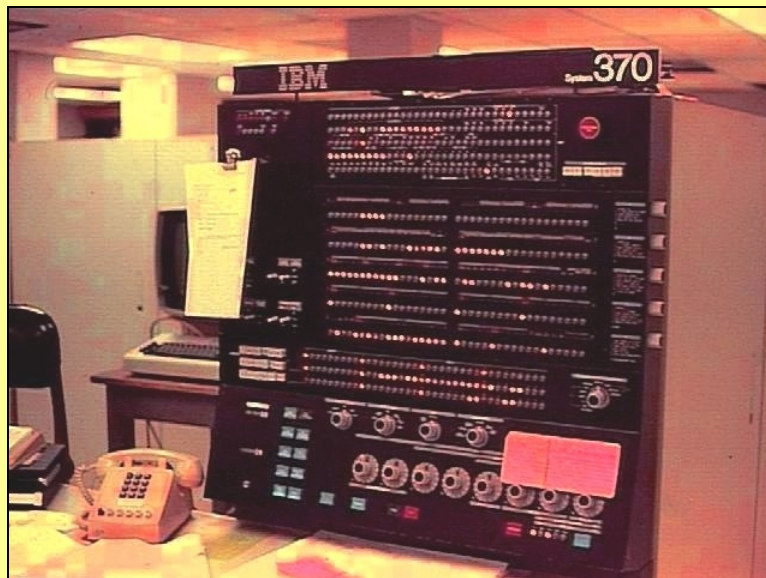


System/3  
Mini



AS/400-B20  
Mini

*Some smaller IBM computers: Fun to use, but not mainframes.*



370/155 - 1970's IBM Mainframe - Big Iron

What is a "mainframe"?

An IBM mainframe computer is a member of the System 360/370 line of systems, and beyond (since 1965), excluding the 360/20 (small and non-standard).

Mainframes are large and powerful computers capable of high-speed number crunching, handling massive amounts of data (I/O) including printing, running dozens of batch jobs at the same time, and handling hundreds or thousands of online users.

Some of the so-called "mainframes" which have been replaced by smaller machines were actually MINI-COMPUTERS - not mainframes.

The IBM 360/20 was really a mini-computer with 8 small, 16-bit registers. The IBM 1130 was a popular, small, fast, scientific computer, but not a mainframe.

The System/3 and the AS/400 are mini-computers.

The IBM 4341 computer (not shown) was always considered to be a small machine, and was closer to a mini-computer than a mainframe.

So, if someone says, "We replaced our 'mainframe' (360/20, System/3, AS/400) with a mini-computer and dozen servers", tell them they did not replace a mainframe: They replaced a MINI-COMPUTER with OTHER SMALL COMPUTERS because their old computer wasn't a mainframe.

Conversely, if a real mainframe is replaced by a LARGE "mini-computer" with multiple processors and huge memories, etc, then the so-called "mini" is close to being a mainframe anyway. They replaced their mainframe with a GIGANTIC "mini-computer". After all, the tallest midget in the world is 17 feet tall. And the biggest so-called "mini-computer" in the world is huge, just as Reno Nevada is "The Biggest Little City in the World", with a population of over a quarter of a million people. And "The Biggest Little Computer in the World" has a quarter of a million little processors, weighs 600 tons, occupies 17 acres of floor space, and sells for \$3,250,000,000 and leases for \$495 million per month plus tax, air conditioning, and nuclear power plant. Maybe it's not really a "Mini".

Occasionally, a large mini-computer may be replaced by a desktop server, but that scenario is possible only if the mini was mostly loafing, and printed only a few dozen pages of output per day.

Even though the smallish IBM computers were far less powerful than mainframes by the standards of today or yesteryear, at least they were reliable and up to the tasks given them.

There's nothing wrong with the smaller computers, and it would be fun to own one and have it running in my basement. But the smaller computers were never considered to be mainframes by those who maintained the big iron.



## 9b. Mainframe Replacement Nightmares

*"Client-Server? Chaos. A nightmare. I've talked to hundreds of companies attempting the transition [from a mainframe to a client-server environment], and only one has fully succeeded."*

Richard Finkelstein, 1994, president of Performance Computing in Chicago.

Apparently, client-server sales people have no idea how to size an application - at least not mainframe applications on client-server machines. Based on the wreckage and huge debts they've left behind, it would appear that they're clueless when it comes to mainframe power, scalability, and to their own machine's administrative requirements, etc.

When it comes to replacing a "real" mainframe with multiple mini's or micro's, the best advice is to forget it: It's been tried many times and failed, costing companies anywhere from \$16 million to \$500 million. In the end, they've had either nothing to show for all their efforts (except huge bills), or terrible response time with HIGHER costs and larger staffs.

Minis and micros simply do not have the power or the scalability of mainframes - not to mention the nightmare of trying to manage 10 or 100 mini-computers instead of 1 mainframe.

In one \$200 million debacle, the cost to run Unix servers rather than the mainframe was double the mainframe cost, and the Unix version did not work. Project status: Abandoned - converting back to the mainframe.

See <http://www.actscorp.com/reboothill.htm> for a dozen horror stories about such conversions. Here's one of their stories:

Project:

Replace the mainframe system with a client/server system (Sequent/Unix) to "lower the costs of computing".

Organization: Motor Vehicle Licensing Agency

Amount invested: \$100 Million Plus

Unanticipated problems:

The application could not support more than 50 users with an 'acceptable' (??) response time of 10-15 seconds. With 100 users online, the response time frequently increased to minutes.

The supplier, scrambling for some type of solution, actually put data integrity at risk by convincing the customer to remove record-locking protection on updates. Their rationale was that "it's a one in a million chance of 2 users accessing the same record simultaneously."

After this change the response time improved to 10-15 seconds for up to 100 users, but still increased rapidly thereafter.

The integrity problem is that, as they issue around 3 million licenses per year, the "one in a million" chance actually occurs about 3 times per year when duplicate licenses are issued to different vehicles.

Project Status: Mainframe and client/server systems kept.

The mainframe was finally 'replaced' some 3 years later than expected, but the user still uses an outsourcing firm to run and maintain their legacy mainframe applications, at a cost close to the original total mainframe cost!

*[...Which means, the mainframe was NEVER replaced - it was simply 'moved offsite' and is still being used by that agency.]*

Final budget: 3 times the old mainframe budget.

Anticipated cost: 6 times the anticipated cost.

In addition there has been a 50% increase in clerical staff to handle the same volume of transactions, adding further to costs.

A terrible and expensive decision, at taxpayer expense, it still doesn't work, and it's still not finished!! Client/server is just a money pit for that state agency.

### 9c. Mainframe Replacement - Wiseguy challenge.

Okay, you Replacement Wiseguys out there: Put your personal funds at risk for an opportunity to make a lot of money - or lose everything.

As the CEO of the fictitious XYZ Corporation, a company that uses IBM mainframes plus PCs and LANs, I'll deposit into your checking account our entire annual mainframe budget, averaged over a 10-year period, every year for the next 10 years.

This covers the costs of hardware, software, maintenance, electricity, heating and air conditioning, floor space, programmers, Tech Support, tape libraries and librarians, admin people, computer operators, training and education, supplies, etc. Everything. All dollars will be converted to today's dollars before averaging, to allow for inflation, and increased annually by the rate of inflation, resulting in more money deposited into your account. Last year's annual budget was \$25 million and the 10-year adjusted average was \$30 million. I will lend you \$100 million to cover conversion expenses, and deduct it 10 years from now.

Out of your checking account I will withdraw the costs of converting to multiple mini-computers, and the costs of running them using the same parameters as above. The entire mainframe budget will also be deducted from your checking account since it's a double expense during the conversion. One-time budget hits (costs of conversion plus new mini-equipment, etc) will be amortized over the remaining life of the challenge.

We will unplug the mainframe and sell it as soon as everything is working satisfactorily on the mini side, producing the same reports (or their online equivalent), on time, with good terminal response time, little or no downtime (meeting or exceeding the standards set by the mainframe), etc. When the mainframe is unplugged, your expenses will be greatly reduced, and you may begin making some big money!

This will continue for 10 years. Every year, I will deposit the same fixed amount (increased by inflation) we have spent on the mainframe, and deduct what we are spending on the mini replacements.

At the end of 10 years, I will deduct the \$100 million loan I made to you.

Penalties: I will charge you for every late report, every incorrect report, every screen displayed with incorrect information, every computer-caused incorrect record, unplanned computer downtime, every second of additional response-time delay to each user's terminal, and every employee who quits due to frustration with the new system. These charges will be based on standard rates involving employee's time, projected lost sales as computed by our marketing department, etc.

There will also be significant penalties for not having a Disaster Recovery Plan in place, at least partially tested, and approved by the auditors, as well as penalties for being unable to upgrade hardware and software, being unable to increase system power as needed, being unable to meet seasonal peak-demand loads with at least half decent response times, being unable to measure and report on system performance, being unable to perform adequate capacity planning, putting data at risk through any means including a lack of proper record and file protection or a lack of memory protection, an inability to audit the files and programs, an inability to produce old or new tapes or information for the IRS, a failure to meet an employee payroll deadline, etc.

If such situations have existed on the mainframe side during the past 10 years, funds will be added to your checking account as MY penalty and your gain, using the same calculations as above.

Outcome: At the end of 10 years, you could be very wealthy, successfully converting from our mainframe to dozens of cheap mini computers, pocketing a huge pile of cash - perhaps tens of millions of dollars! On the other hand, if you owe us money and can't pay, or if you try to leave the project or leave the country, your cars and homes will be confiscated, and you and your family and friends, the minicomputer sales people, and all the mini employees you hired, will be - you know - part of "Mainframe Replacement History", sleeping with the mice in a low-rent location, unable to afford anything habitable, with no cars, no passports, and no money...  
Why should your company take all the risk??

## 10. Viruses, etc.

```
//MYTEST    JOB 'IEFBR14 - I LUV U',CLASS=T,REGION=8K,TIME=(,2)
//STEP1     EXEC PGM=IEFBR14X    <==== MY CLEVER VIRUS PROGRAM
//TESTFILE  DD   DSN=MYFILE.TEST,DISP=(OLD,KEEP,KEEP)
//**
//**  this is yet another test to see if I can somehow trick mvs
//**  into swallowing this clever virus and erase all the files
//**  on the hard drives
//**  this is test number 14,172,484
//**  the last 14,172,483 tests failed but im hopeful that somehow
//**  i can maybe cause some kind of buffer overflow or run a script
//**  or maybe lower its shields or maybe be phooled by the
//**  programmers name field or bypass RACF or something...
//**
//**  ping jason - what am i doing wrong and what is a RACF?????
//**  i cant phool it
//
```

Mainframes do not catch viruses.

A mainframe does not need to "lower its shields" to scan a job which needs to be run, or allow a user to logon.

It has no "medium security" setting, and you can't place a "cookie" in a production file - even if you wanna.

Operators and programmers do not randomly execute unknown programs "just to see what will happen", and if they did, the job would probably fail anyway.

You never receive messages which appear to be from a colleague, but are actually from someone else unknown to you.

You never receive notifications that a Job which you didn't submit, and don't know anything about, had a JCL error (it could be done on a very limited scale, but why bother?), etc.

You cannot infect a mainframe with JCL or data, and JCL does not contain a secret script file, or cause a buffer overflow which somehow the mainframe accepts as unimportant, then proceeds to execute the overflowed data as if it were a program, while security people issue worldwide warnings to be on the lookout for jobs named

"IEFBR14 - I LUV U"

and all its possible permutations, such as:

```
"IEFBR14 - I LUVV U 2"...
"IEFBR14 - I LUVVV U T0000"...
```

Mainframes do not run "not ready for prime time" software that allows such nonsense and has caused so much damage and loss of sleep in the PC world.

IBM did a terrific job in designing and implementing computer security.

## 11. Trying it Again.

If a batch job didn't work the first time, it won't work the second time either. There's no point in "trying it again" - unless the first job fixed something which a preceding step should have taken care of, or someone freed some datasets they shouldn't have been holding. Usually, that would mean you have a problem with your JCL or scheduling. If it goes into Production like that, it won't work then just as it didn't during testing.

## 12. Maturity and purpose.

IBM mainframe software was written by adults for businesses, academia, scientific research organizations, and government institutions. It was written with thoughtfulness, care, consistency, and purpose for intelligent users.

Its terminology is, for the most part, meaningful and dictionary-based. Its documentation has the formal foundation of the English language, and was written by educated people who understand what words mean, could write a proper declarative sentence with a verb, a noun, a subject and an object, and could usually explain a concept or an action with clarity.

[Compare with Microsoft's "INVISIBLE BASIC" (Visual BASIC) worthless documentation in its language specifications manual. Visual BASIC and its worthless manuals is just useless junk written by Third Graders. Unbelievable.]

It is in use by small companies, and by large ones, with billions of dollars at stake in its reliability, accuracy, and documentation. The same is true of the hardware. They are in use by the FAA to track airline flights, banks, insurance companies, the Pentagon, hotel chains, electric companies, oil companies, etc.

The product of IBM's work is something which can be truly admired.

### 13. VTOCs and Catalogs.

#### Mainframes:

A VTOC (Volume Table Of Contents) resides on disk, and describes the contents of that disk volume. The files are present on that disk volume. They may or may not be cataloged.

A catalog contains entries for the names of files, what kind of medium they're on (tape, disk), and the volume serial number(s) of the volume(s) (VOLSERs). The files may or may not actually be present. The catalog is a good thing.

Example of JCL for locating and using a non-cataloged disk file:

```
//FILE1 DD DSN=MYFILE.TEST,DISP=OLD,
// UNIT=SYSDA,VOL=SER=771234 <==== for locating the file
```

Example of JCL for locating and using a cataloged file on disk or tape:

```
//FILE1 DD DSN=MYFILE.TEST,DISP=OLD <==== look in the catalog
```

It is **\*\*much easier\*\*** if everything residing on disk is also cataloged, to keep from getting confused, or reading the wrong file.

Tape and disk files can be cataloged and uncataloged without disturbing or even needing the files involved.

#### PCs under Windows:

No Windows equivalent for the catalog. The closest Windows comes to a catalog is the Registry.

The Windows equivalent for the VTOC is the FAT (File Allocation Table) or similar entity for NTFS volumes, etc.

### 14. Partitioned Datasets (PDS's).

A Partitioned Dataset is essentially the same thing as a "Folder" in the PC world, but only 1 level deep. IE, a PDS does not contain other PDS's. The members in a PDS are the equivalent of "Files" in the PC world.

All the members in the PDS must have the same attributes (same format, same record length (except for format U), same blocking factor, etc. In other words, it's not possible to mix executable programs with text files, for example, since an executable program has an Undefined record format (RECFM=U), while a text file would normally have a fixed record length and be "blocked". For example:

DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=800) - or whatever someone specified.

Examples of PDS's: SYS1.LINKLIB, SYS1.PARMLIB, SYS1.SAMPLIB.

Examples of PC Folders: Windows, Windows\System32, Program Files.

Mainframes	PC's
-----	-----
PDS	Folder/Directory
Members	Files

## 15. Ease of use.

PCs are certainly easier to use, in most respects...  
No argument there. And they're rapidly catching up in speed.

If I want to backup my 16MB MVS Sysres pack on my old "2001" PC, I just do a copy and paste, disk to disk. It takes about 15 seconds, from start to finish (clicking, copying, locating, pasting, etc) with only about 1 or 2 seconds(?) needed for the actual copy.

No JCL, no tape mounts, no TYPRUN=HOLD, no calling the computer room, no JCL errors...

Of course, my little Sysres is only 16 megabytes as opposed to hundreds of megabytes in a real shop, on a 3380 or 3390, but still...

An old 3330-1 disk drive from the 1970s had a capacity of 100MB. The time to copy a full 3330-1 volume would have ranged from about 2.3 minutes (full track reads) to 7 minutes (normal software, 2 channels, 30% channel busy), depending on software used and machine availability, according to my calculations. If I copy a 100MB file to another physical drive using Windows' copy and paste, the operation takes about 5 seconds, giving a thruput rate of 40MB/sec (100MB read + 100MB written = 200MB. 200MB/5secs = 40MB/sec total thruput, or 20MB/sec reading + 20MB/sec writing). Not bad!

(I've seen disk-to-disk thruput rates of 10-40MB/sec on my older PC, using 2 physical hard drives. This is not the same as "disk speed".)

Again, no JCL, no tape mounts, no TYPRUN=HOLD, no calling the computer room, no JCL errors... That can be a major time savings!

Comparing the disk speed of a fairly modern PC to a 30-year old mainframe disk drive is simply comparing technical progress over a 30-year period. The fact that an F-86 Sabre jet (1950) is faster than a WWI, 4-engine, Russian bomber with a crew of 7 (Ilya Mourometz - 1914) - just as an F-16 is faster than a B-36 - is hardly surprising.

But the convenience - the ease of use - of the PC, is terrific and is partly what caused sales to skyrocket (not to mention the aspect of "owning" your own computer!!).

Newer mainframe disk drives - 3390s and above - are much faster than 30-year old 3330's and 40-year old 2314's, of course. And the new ESCON and FICON channels (fiber optics) are unbelievably fast!

And the fact that the mainframe can have dozens of disk and tape drives and printers working hard simultaneously at high speed, keeps the mainframe as the king of "total power". If Microsoft, AOL, and some others could just produce decent, non-bloated, non-stupidly written software for Windows, MS/Word, Wordpad, Internet Explorer, and some other programs, they'd really have something.

"Copy and Paste" is about the only feature that works really fast on my PC. Everything else (except MVS under Hercules) ran like an overloaded WWI Russian bomber in a headwind during a blizzard, under Windows/XP Pro SP2 - until I upgraded to a faster CPU, more CPU's, far more memory, and larger disk drives (for less file fragmentation). That's a lot of money and hardware just to serve one person's computing needs...

But with that "ease of use" feature, life is convenient and often satisfying with a fast Personal Computer and a Personal Mainframe simulator. The mainframe simulator doesn't simulate everything, but I can live without occasional trips into the noisy and ultra-cold "real" computer room, kept far colder than necessary (according to our IBM Customer Engineer)...



## 16. Cost.

Although prices have dropped, mainframes are still expensive. If I remember correctly, our company paid around \$2 million in 1980 for a low-end, used 370/158-3 mainframe. It had only 4 megabytes of memory and only 4 I/O channels. Still, we ran a 500-store company with it, and supported about 30 online programmers using SPF.

## 17. The IBM z/Series mainframes.

A Recent IBM z/Series System: The z990 (described in 2003).

Condensed from IBM's website:

z990 Models: 4

Architecture: 64-bit.

Optional compatibility architecture per LPAR: 24-bit and 31-bit.

Max Processors (internal CPUs): 32.

Max I/O channels: 1024 ESCON or 240 FICON-Express-2 (full duplex).

Channel speeds: Hundreds of megabytes/second (very fast).

Max memory: 256 Gigabytes.

Max unique system images (LPAR's): 30 (TEST, PROD, TECH, DB, etc).

Max MIPS: 9,000 (9 billion instructions/second - not megahertz).

[Note: MIPS are not very meaningful, but I listed them anyway.

Some mainframe instructions are the equivalent of a subroutine, performing a great deal of work.

What matters is the throughput rate, and mainframes such as the z990 have awesome throughput rates].

Supported Operating Systems: z/OS, z/VM, OS/390, (z/VSE?), and others.

Max e-transactions per day: 450 million.

Max e-transactions per day as a clustered z990: 13 billion!!

Brief summary of "max MIPS" of some older processors and recent IBM Z-Series processors:

Max MIPS	Instructions/sec	IBM Computer Model
-----	-----	-----
Old---		
.09	90,000	360/40 (1960's)
1	1 million	370/158-3 (early 1970's)
13	13 million	3081-K
26	26 million	3084-QX, 4 CP's, 48 channels max
102	102 million	3090-600S
846	846 million	9672-R96 G5
1,644	1.6 billion	9672-ZZ7 G6
New--- (Z-Series)		
9,000	9 billion	z990, 32 CP's (2003)
18,000	18 billion	z9-109-S54, 54 CP's
34,560	34.5 billion	z10, 64 CP's, 1 frame (approx)

MIPS are only part of the story, but the numbers are quite impressive.

Note, for example, that the old 3084-QX, from the 1980's, could have up to 48 channels. That's a max of 48 I/O operations occurring simultaneously (geometry permitting), while the CPU's worked on other things. It's ALMOST like having 52 processors (48 + 4).

## 18. Confessions.

This document was produced on a PC running a Microsoft OS.

It's physically small, it's fun, and it's mine!

Logically, it's a BIG system - bigger than any of the early mainframe systems I worked on in terms of speed, disk space, and memory.

But 40 years ago, PC's this size and this powerful were not available.

And I've applied security fixes every week for 20 years...

Hundreds!!!

IBM mainframe operating systems were MUCH better at security than Microsoft has ever been!!

But it's fun, and it's mine.

.....

#### Acknowledgements

Thanks to many people for providing additional information, and pointing out technical errors in this document. I'm very grateful!!

I don't remember all the names, but some of those people and companies are:

IBM Corporation  
Roger Bowler  
Jay Maynard  
Volker Bandke  
Melinda Varian  
Ron Tatum  
Mark Waterbury  
Rick Fochtman  
somitcw  
Bob Abeles  
Ivan Warren  
Sam Bass  
John Klensin  
Seymour Metz  
Jeff Ogden  
George Davis  
Stuart Linderman  
David Stuart  
Peter Schaeffer

#### Credits

Some photos are copyright IBM Corporation.

---Comments and corrections are appreciated---

It's important to me that this document is 100% accurate.  
If you notice any errors, please feel free to contact me and point them out. Thank you.

Omissions: I didn't and couldn't include EVERYTHING.  
But if you think I've omitted something important that can be covered in a few sentences (maybe more than a few sentences), please let me know, and I'll add it if I can.

When writing to my email address, please be advised that I can send only HTML email via AOL. The option to send plain-text email was dropped by AOL several years ago.

Dave Morton.  
[marspyrs@aol.com](mailto:marspyrs@aol.com)

---End---