

What is a System i?

By Dr. Frank G. Soltis

Although most of the world's population would have difficulty answering the question "What is a System i?" it should be a fairly straightforward question for those of us familiar with this server. Or is it? We may live with an AS/400, iSeries, or System i every day, but when it comes to defining exactly what it is, many of us may stumble a bit.

It's an IBM eServer. It's a midrange server. It's a business computer. It's used by businesses all over the world. It's sold in more countries than McDonald's hamburgers. It has the largest number of customers of any IBM server. It's all of this and more, but we still don't have a good definition of a System i and what makes it unique among servers.

With the movement to converged hardware among IBM servers and running the AIX operating system on System i, the definition of a System i has become increasingly blurred in some people's eyes. The purpose of this article is to try and clear up some of this blurred vision.

The Problem with Definitions

At the time of the model i890 announcement, I decided to write a whitepaper entitled "What Is a Mainframe?" The model i890 is clearly a mainframe-class server, and at the time we were thinking about simply calling it a mainframe. But you may have noticed that today IBM does not call the model i890 a mainframe, at least not in any official publications. The reason is that within IBM we could not agree on what was a mainframe.

On the surface, it seems that finding a definition for a mainframe should have been easy. After all, IBM has been building mainframes for almost 40 years. My search for a general definition of a mainframe quickly showed that finding a good definition was not going to be easy. To be sure, there were many different definitions, but they all seemed to fall into two distinct categories: broad general definitions and definitions that concentrated on some particular aspect of mainframe computing. Neither of these categories, unfortunately, provided a very good definition.

The general definitions of a mainframe most often describe only the size of the computer. For example, a mainframe is sometimes defined as a room or more of computer equipment. This definition also fits a PC server farm, and few would argue that a PC server farm is a mainframe.



Léo Lefebvre

Dr. Frank Soltis in a "Back to the Future" skit at the 2007 COMMON Annual Conference in Anaheim

Another widely used general definition describes a mainframe as a large computer for the commercial applications of Fortune 1000 businesses and other large-scale computing purposes. This definition also suffers from too many ambiguities. By this definition, any large computer that is used by a Fortune 1000 business to run commercial applications is a mainframe, and we know that is not true either.

The more specific definitions of mainframe usually get too specific. They usually define a mainframe by the particular operating systems it runs. If it runs zOS, MVS/ESA (OS/390), VM/ESA, or VSE/ESA, it is a mainframe. Some of these definitions go so far as to define a mainframe as the operating System itself. Linux running on a mainframe totally confuses these definitions. If Linux is a mainframe operating system, then any computer that runs Linux must be a mainframe. Again, not true.

It should be fairly clear by now that there probably is no good definition for a mainframe that satisfies everyone. A few years back, there was a joke circulating through the various S/390 user groups around the world that has become a personal favorite of mine for the definition of a mainframe: If you can pick it up, it's a PC. If you can't pick it up, but you can push it over, it's a midrange server. But when you can't pick it up or knock it over, it's a mainframe.

At 6 feet 8 inches tall and weighing almost a ton, you are definitely not going to push over an i890. By this definition, it is a mainframe. But since few others in IBM thought this was a good definition, we decided not to call the i890 a mainframe. We also never published my whitepaper. ➔



The System i 595 is currently the largest System in the System i family. It offers multiplatform management and up to 64-way symmetrical multiprocessing capability. It can support 2 TB of memory, 381 TB of disk, and up to 216,000 CPW, but is it called a mainframe?

Finding a good definition for the System i entails all of the same problems as finding a good definition for a mainframe. We could simply say that a System i is any server that runs OS/400, but is there more to it than that?

It's Not the Hardware

One way to determine what a System i is, is to look at what it is not. For example, it is not a hardware box. Oh, IBM does sell you a black box, and on the front of that box, it does say "IBM System i," but the hardware is just the visible part of the System i. The System i itself is the virtual machine that resides inside the black box.

The System i has never been defined by its hardware. It is defined by software. This is in stark contrast to every other commercially available server, where the hardware architecture is the defining element of the server. This means that all software in these servers, including applications, is dependent on the underlying hardware architecture.

The power of not defining the server by its hardware was dramatically illustrated when the AS/400 processor technology moved from



Vaughn Dragland

Looking inside a System i

Modern RPG & Cobol Web Applications

IceBreak Application Server for System i

- ☑ Build new Web applications and convert existing green-screen applications
- ☑ Does not require any additional hardware or software including CGI, Java, Apache, WebSphere, WebFacing
- ☑ Supports XML, Web Services, SOA, AJAX, SQL
- ☑ Deliver Web applications that your users and customers will say "Cool" & "Wow"
- ☑ Very EASY learning curve
- ☑ RPG business logic is developed and maintained separate from the XHTML user interface

User Group Members

- ☑ Try IceBreak completely FREE for 30 days
- ☑ No Gain — No Pain. Let us develop a Web application for you and you pay only if you purchase IceBreak.
- ☑ Receive a 30% User Group discount off the IceBreak Enterprise Edition (Limited time User Group offer)

For additional information or online demo, contact

www.IceBreak4RPG.com

System & Method International

info@IceBreak4RPG.com • 1-888-290-3256

www.IceBreak4RPG.com • www.system-method.com

CISC to RISC in the middle 1990s. Customers saved the programs from their CISC machines, restored them on their new RISC machines, and the programs ran as full 64-bit applications without even having to be recompiled. No other server has ever made such a major hardware transition. Those that have tried failed. For example, Digital in the 1990s tried to move its VAX installed customers to RISC hardware but was not successful. Today others, including HP, are just beginning to make the transition to a new hardware base, and their success is by no means assured.

We used to talk about this design concept of a virtual machine as technology independence. In recent years, the term "virtualization" is being used more frequently to describe the use of software rather than hardware to define the interfaces to various parts of the server.

The point is that the System i is not defined by its hardware. That hardware can in the future change completely, and we will still have a System i. If it is not the hardware that defines the System i, it must be the software.



It's Not the Operating System

A few years back, there was a great deal of discussion in the industry about operating systems not being very important. Even though not everyone agreed, there were those who predicted that all proprietary operating systems would eventually go away. That hasn't happened yet, but there are still some people who believe it will.

Defining a server in terms of its operating system would seem to be a short-lived definition, especially if you believe that all operating systems are similar and that it is unimportant which one you choose. Without arguing for or against either side of this discussion, it is true that all operating systems do pretty much the same things.

Any computer science student can probably tell you that an operating system fundamentally does only two things. It controls all of the computer's resources, and it provides the base upon which

applications are written. That student can also probably tell you that an operating system has four major components: process management, memory management, input/output, and the file system. On the surface, it seems hard to believe that one server can be distinguished from another solely on the basis of its operating system. This gives credence to the argument that one operating system is as good as another.

Notice also that the components of an operating system deal primarily with the underlying hardware. An operating system must know the details of the hardware. In other words, an operating system cannot be technology independent. Moving an operating system from one hardware platform to another requires a great deal of work, assuming you want to have an efficient implementation of that operating system. For example, it took nearly five years and hundreds of system programmers to rewrite the operating system when the AS/400 moved from CISC to RISC.

Because the operating system is dependent on the underlying hardware and must be heavily modified to run on different hardware, the System i is not defined by its operating system. If we were to change the operating system of the System i completely, it would still be a System i.

A Poor Choice for a Name

Now here's where it gets tricky. What is the operating system of the System i? If you said i5/OS (or OS/400), you would be wrong. None of those four major components of an operating system described in the previous section are in i5/OS. Simply stated, i5/OS is not an operating system, and it never has been.

Twenty nine years ago in October 1978, the IBM System/38 was announced. That system was the beginning of what has become today's System i. Because the S/38 was technology independent (defined by its software and not by its hardware), any component of that system that was hardware dependent was packaged as a part

UPGRADE YOUR IMAGE

The PENTAX Machstream 32 continuous form laser printer has 600 dpi print quality. At 32 ppm and MICR enabled, the Machstream 32 has the reliability, durability and usability expected of a production printer.

All at a line printer price point!

MACH^{PENTAX} CONTINUOUS FORM LASER PRINTER **32**



pentaxtech.com • 1-800-877-0155 x 1615 • laserinfo@pentax.com

PENTAX[®]

Profile on Dr. Frank Soltis

Dr. Frank Soltis holds the title of IBM System i Chief Scientist. He is the creator of the technology-independent computer architecture and the single-level addressability that led to a totally new breed of IBM computers, including the AS/400, the iSeries and the System i. He participated in the definition of the Power processor architecture and led the effort to move IBM systems to the 64-bit Power processors. Variations of these Power processors are now used in consumer electronic devices; for example game consoles from Sony, Microsoft and Nintendo, and in the world's most powerful supercomputers.

The author of *Inside the AS/400* and *Fortress Rochester, The Inside Story of the IBM iSeries*, Dr. Soltis is a frequent speaker at computer conferences around the world. In addition to his responsibilities as an IBM Chief Scientist, he is an Adjunct Professor in the Department of Electrical and Computer Engineering at the University of Minnesota where he teaches graduate courses on advanced computer architectures.

Currently based at the IBM Development Laboratory in Rochester, Minnesota, he travels the world speaking on IT trends and technology advancements. In addition to his several award-winning books, Dr. Soltis has numerous technical papers and other publications to his credit. He holds more than 25 patents and published invention disclosures related to computer systems. In his spare time he enjoys working on and racing Porsches with his sons.

of the hardware. As a result, the operating system of the S/38 was packaged as part of the microcode. Microcode in those days was considered to be hardware.

Other system software that was not hardware dependent was packaged separately. This software package included the world's first commercially available relational database, functions to manage work in the system, a new control language, data definition interfaces, extensive security and authorization facilities, and some powerful program development tools. Because many of the functions in this system software package had to do with the overall control of the S/38, we decided to call the package Control Program Facility (CPF). CPF was not an operating System in the textbook sense, although it was sometimes called an operating system for lack of a better, more descriptive name. The real operating system of the S/38 was in the microcode.

In 1988, the S/38 was renamed the AS/400 — yes, we have renamed this system more than once. Someone, obviously a marketing type, also decided to rename CPF to OS/400. In hindsight, this was a huge mistake. Suddenly, CPF, with all its sophisticated functions and capabilities, was degraded to the level of a simple operating system. To this day, people still try to compare OS/400 to a Unix or Windows operating system. That's like comparing an aircraft carrier to a rowboat. They both float, but similarities end there.

When we again renamed this system from AS/400 to iSeries, some of us wanted to change the name of OS/400 to show that it was far more than just a simple operating system. That name change was never made because the OS/400 name was the only link that customers, many of whom weren't too happy about the name change anyway, had to the AS/400.


Still another change made in 1988 for the AS/400 was to drop the name microcode and instead use the name licensed internal code (LIC) to describe the system software that was

dependent on the hardware. Microcode is part of the hardware, and customers can own the hardware. System software, including operating systems, is generally treated as licensed software. The name change meant that the AS/400 operating system, which had previously been considered hardware, became licensed software. With the introduction of the RISC systems, that licensed software became System Licensed Internal Code (SLIC). SLIC is the operating system of the System i.

i5/OS: The Defining Entity

There you have it. The one thing that made an iSeries an iSeries was OS/400, and what now makes a System i5 a System i5 is i5/OS. The specific hardware isn't important. It has and will continue to change in the future as more and more common components are shared across IBM's servers. SLIC, the operating system, is also not the big differentiator. It has undergone major changes over time and will change even more in the future as system software components are also shared among IBM servers.

i5/OS, on the other hand, is unique to the System i5. It is i5/OS that provides the control environment and the application environment that make the System i something very special. It is not the simple operating system that its name implies. Maybe someday in the future we will have the opportunity to give i5/OS a name that properly reflects its true value.

Meanwhile, the next time someone tries to tell you that in the future you will be able to buy a System i5 without i5/OS, set the record straight. Without i5/OS, it is not a System i. It's just another server. 

Dr. Frank G. Soltis of IBM Rochester created the technology-independent architecture used in the AS/400 and System i. He is IBM's System i chief scientist and a professor of computer engineering at the University of Minnesota.

This article is reprinted with permission from iSeries NEWS, September 2003, article ID 17055, www.SystemiNetwork.com. It was updated slightly to reflect new system names.

