

What on Earth is a Mainframe?

An Introduction to IBM Z Mainframes and z/OS Operating System for Total Beginners

Free sample chapter courtesy of Longpela Expertise.

But the book at www.longpelaexpertise.com.au/books.php

David Stephens

Long *pela*

What On Earth is a Mainframe

An Introduction to IBM Z Mainframes and z/OS Operating System
for Total Beginners.

Second Edition.

Copyright © 2008-2025 by David Stephens

All rights reserved.

First Edition: October 2008

Second Edition: June 2025

ISBN 978-1-326-53271-0

Order this book online at www.longpelaexpertise.com.au/books

Produced by Longpela Expertise

Longpela Expertise provides Mainframe systems consulting services.
Find out more at www.longpelaexpertise.com.au

Acknowledgements

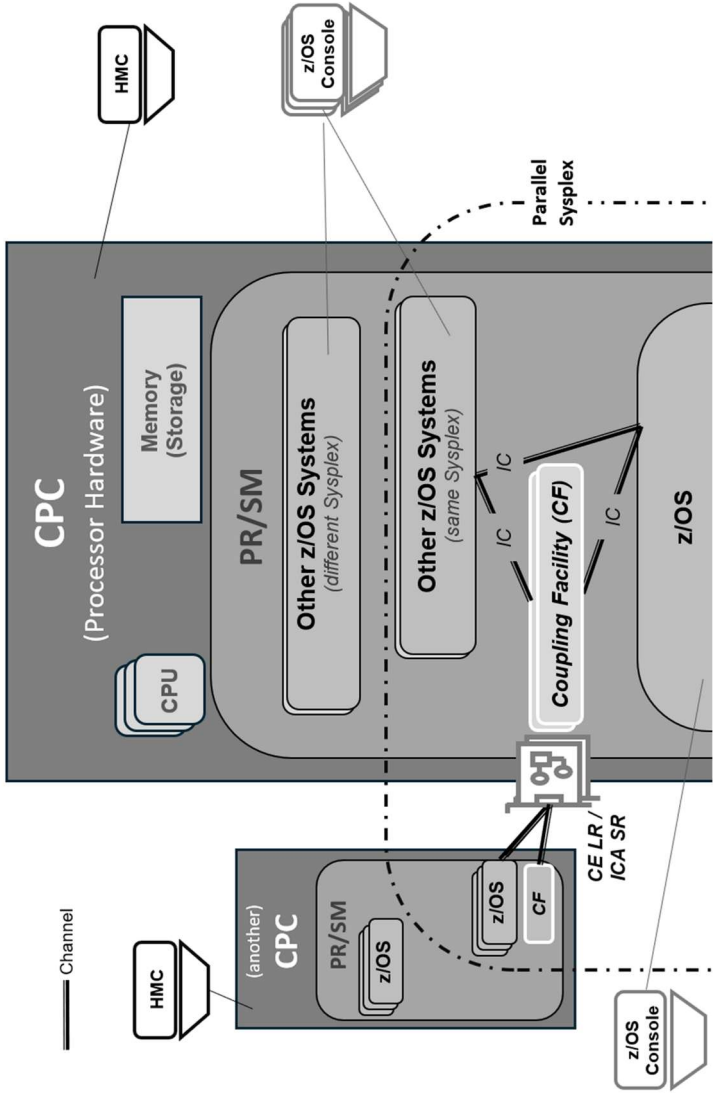
You can't write a book without help from a lot of people. So here they are:

- Paul Norris, Andrew Binnie, Hecate Jay, Mark Beckham, Brian Breathnach, Dave Matoe and Louise Wilson for their help in the first edition.
- vintagecomputer.ca for supplying images.
- All who left reviews on Amazon and other sales sites: I read these when preparing this second edition.

And special thanks to my wife Simone, who puts up with me babbling on and on (and on) about this book. She is also the reason why the Mainframe map exists, and why the cover looks so good.

The Mainframe Map

Here's a simplified diagram of a Mainframe. It may not mean much now, but use it as your map as you travel through this book.



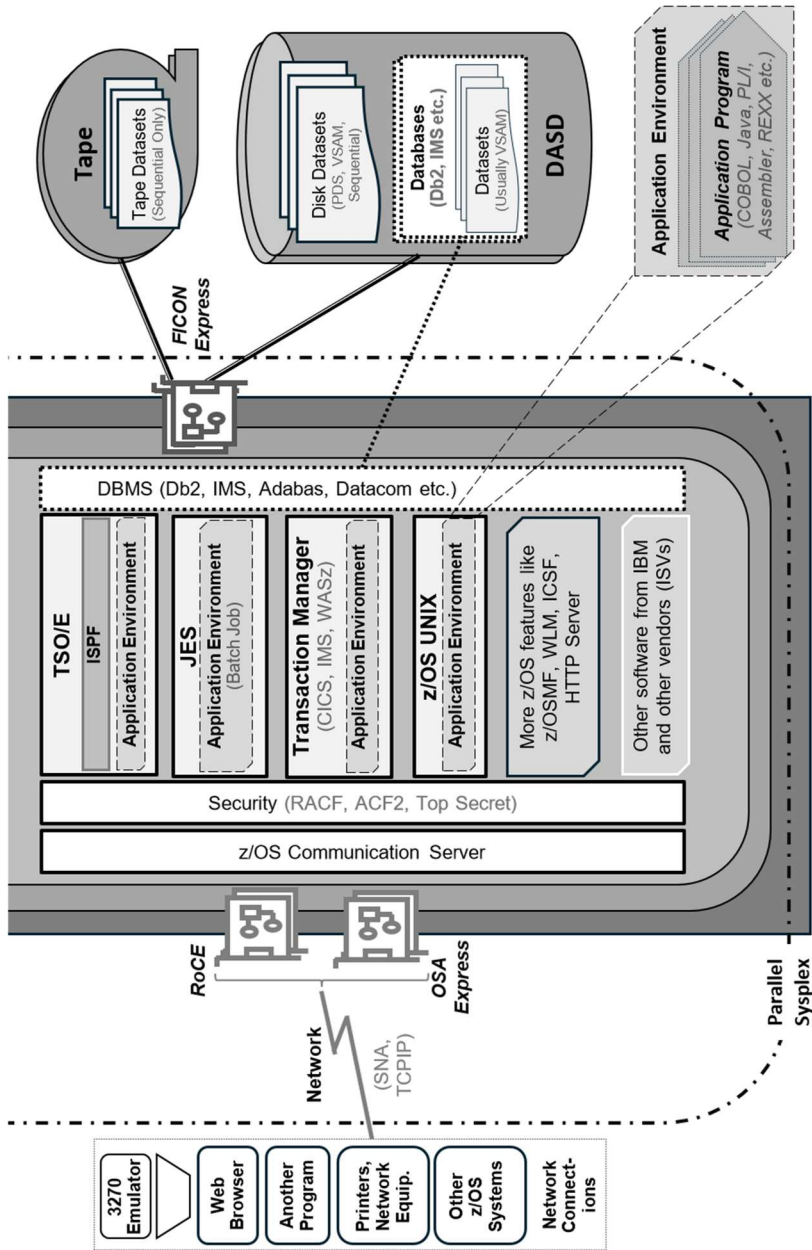


Figure 1: A Mainframe Map

Contents

| | |
|--|-----------|
| The Mainframe Map | 4 |
| Contents | 7 |
| Introduction | 11 |
| Chapter 1: But Aren't Mainframes Dead? An Introduction to the Mainframe | 13 |
| What is a Mainframe? | 13 |
| If Mainframes Are So Good, Why Does Everyone Want to Get Rid of Them? | 16 |
| Why Keep Mainframes? | 18 |
| The Last Word | 22 |
| Chapter 2: What Are All These Refrigerators? Mainframe Hardware | 23 |
| The Brains of the Organization - The CPC | 24 |
| Why Can't They Just Call Them Disks? | 30 |
| You mean Mainframes are still using TAPE? | 32 |
| Connecting the Boxes – Channels | 36 |
| Printers | 39 |
| A Special Room for your Mainframe | 39 |
| The Last Word | 41 |
| Chapter 3: The z/OS Jigsaw Puzzle - The Many Pieces of z/OS | 43 |
| What You Get When You Buy z/OS | 43 |
| 12 Things You Need To Know About z/OS | 45 |
| What You Don't Get When You Buy z/OS | 56 |
| The Last Word | 63 |

Chapter 4: Talking to z/OS - Networks and Communications 65

| | |
|------------------------------------|----|
| Network Type 1: SNA _____ | 65 |
| Network Type 2: TCP/IP _____ | 72 |
| Talking Between Applications _____ | 75 |
| The Last Word _____ | 80 |

Chapter 5: Putting the Mainframe to Work - Transaction Managers and DBMS 81

| | |
|--|-----|
| The Database Management System – Your Database Nanny _____ | 81 |
| The Transaction Manager – Crowd Management _____ | 87 |
| Transaction Managers and DBMS on z/OS _____ | 91 |
| Other Transaction and Database Managers _____ | 108 |
| Do I Really Need to Buy This Stuff? _____ | 108 |
| The Last Word _____ | 109 |

Chapter 6: Application Development on z/OS 111

| | |
|--|-----|
| What's an Application? _____ | 111 |
| What's an Application Environment? _____ | 112 |
| How to Create an Application Program _____ | 112 |
| How to Develop an Application _____ | 116 |
| Supporting an Existing Application _____ | 120 |
| Why Your Source Code is Important _____ | 122 |
| Why Is It So Hard To Develop Mainframe Applications? _____ | 123 |
| Application Development Gadgets and Goodies _____ | 124 |
| The Last Word _____ | 126 |

Chapter 7: Accessorize - Software for the Mainframe 127

| | |
|----------------------------------|-----|
| Monitoring Tools _____ | 127 |
| Tape Management Software _____ | 128 |
| Reporting Tools _____ | 129 |
| Output Management Software _____ | 129 |
| Software for a Disaster _____ | 130 |
| Performance Software _____ | 131 |

| | |
|-----------------------------------|-----|
| Decision Support Software _____ | 131 |
| File Transfer Software _____ | 132 |
| Printing Software _____ | 132 |
| Other Goodies on the Market _____ | 133 |
| The Last Word _____ | 133 |

Chapter 8: What Do All These People DO? People You Need to Run a Mainframe _____ 135

| | |
|-------------------------------|-----|
| Operators _____ | 135 |
| Operations Analysts _____ | 138 |
| Security Administrators _____ | 139 |
| Application Developers _____ | 140 |
| Application Testers _____ | 141 |
| Database Administrators _____ | 142 |
| Systems Programmers _____ | 143 |
| Other People _____ | 146 |
| The Last Word _____ | 147 |

Chapter 9: What Do We Do With It? _____ 149

| | |
|--------------------------|-----|
| Option 1: Replace _____ | 149 |
| Option 2: Rebuild _____ | 150 |
| Option 3: Refactor _____ | 150 |
| Option 4: Rehost _____ | 151 |
| Option 5: Retire _____ | 151 |
| Option 6: Retain _____ | 152 |
| The Last Word _____ | 153 |

Chapter 10: Mainframe Manager Nightmares _____ 155

| | |
|--|-----|
| What Do You Mean It's Unavailable? _____ | 155 |
| Where Does All the Money Go? _____ | 159 |
| Who's Paying for All of This? _____ | 159 |
| Software and Hardware Support _____ | 161 |
| Being Ready for a Disaster _____ | 161 |

| | |
|--|------------|
| Mainframe People – Where Do I Get One? _____ | 163 |
| Paying Someone Else to Do It _____ | 164 |
| Why Is It Taking So Long? _____ | 165 |
| When Do We Get a New One? _____ | 165 |
| Providing the Mainframe With a Good Home _____ | 167 |
| Keeping it Safe _____ | 167 |
| The Last Word _____ | 168 |
| Chapter 11: The Last Word _____ | 169 |
| Appendix 1: An Executive Summary - The Whole Book in Three and a Half Pages _____ | 171 |
| Appendix 2: Where to Go Next _____ | 175 |
| IBM Resources _____ | 175 |
| Other Information on the Web _____ | 176 |
| Magazines and Articles _____ | 176 |
| Courses, Conferences and Training _____ | 176 |
| Appendix 3: Glossary - Translating the Mainframe Speak _____ | 177 |
| Notices _____ | 195 |
| Index _____ | 199 |
| Table of Figures _____ | 206 |
| About the Author _____ | 207 |

Chapter 1: But Aren't Mainframes Dead? An Introduction to the Mainframe

So, what exactly are these Mainframes that everyone's talking about? Let's find out...

What is a Mainframe?

It's a big computer. In the 1970s and 1980s when almost all computers were big, the term *Mainframe* was used to refer to a few different computer systems. Today most of these are gone, and *Mainframe* almost always refers to IBM's Z computers. And that's what this book is all about. From now on when we talk about Mainframes, we're talking about the IBM Z.

IBM Z computers are amongst the largest computers sold today, and they're used for *commercial data processing*. By commercial data processing we're really talking about database-based applications; putting a piece of data into a database, looking at it, and taking it out. Now this sounds ridiculously simple, but it's not. When you're running big database applications, you need to think about:

- **Data integrity:** My data HAS to be right – all the time.
- **Throughput:** I need to run lots and lots of work – now.
- **Response:** I want the answers NOW – not in one minute's time.

- **Disaster Recovery:** If a disaster strikes, I want to be back up and running. Fast.
- **Usability:** It must do what I want, when I want – oh yeah, and make it easy to use.
- **Reliability:** Don't tell me there's a computer problem – I want it available when I need it.
- **Audit:** I need to be able to find out who's done what in the past.
- **Security:** Only those who should, can.

So now commercial data processing is starting to sound a bit more complicated, isn't it?

Today every business wanting to use computers commercially faces these issues. The individual needs, size, and business complexity will be different; but these basic issues will be there. And there are lots of computing systems and software around that can help with all this. So why do people move to a Mainframe?

Simple: they don't. There aren't many organizations that will install a Mainframe from scratch (unless they're installing Linux on the IBM Z), and we'll discuss why soon. But the fact is that almost all Mainframe users have been using Mainframes for years (well, decades actually). People often call them *legacy* systems.

The bottom line is that Mainframes do commercial data processing well. But there are things they're not good at. Things like:

- **Number crunching.** You won't find many scientific applications for the IBM Z. There are cheaper, easier options out there.
- **Graphics and Geographical Information Systems.** No games like Fortnite running on Mainframes.
- **Look good.** Let's talk about this for a second. You're familiar with Microsoft Windows; the mouse and the nice, fancy screens right? Or even better, Apple iPads where your finger can do magic. Well Mainframes have none of that. To talk to a Mainframe you need a 3270 terminal (a *dumb* terminal – no

processing is done here), or a PC running software that *pretends* to be a dumb terminal (3270 *emulation software*).

These dumb terminals are *character based* - have a look at Figure 1. No windows, no graphics (well, you can, but almost no-one does) - just text. People often call them *green screens*, because in the old days they had green writing on a black background. There are certainly applications connecting to the Mainframe that look good – but you’ll find that all the nice, pretty stuff is done on a different computing platform.

```
-----
Menu Utilities Compilers Options Status Help
-----
                                ISPF Primary Option Menu
Option ==>
0 Settings      Terminal and user parameters      User ID . : DZS
1 View          Display source data or listings     Time. . . : 20:27
2 Edit          Create or change source data        Terminal. : 3278
3 Utilities     Perform utility functions          Screen. . : 1
4 Foreground   Interactive language processing    Language. : ENGLISH
5 Batch        Submit job for language processing  Appl ID . : ISR
6 Command      Enter TSO or workstation commands  TSO logon : SPFFPROCL
7 Dialog Test  Perform dialog testing             TSO prefix: DZS
8 LM Facility  Library administrator functions    System ID : MWS1
9 IBM Products IBM program development products  MVS acct. : FB3
10 SCLM       SW Configuration Library Manager   Release . : ISPF 8.1
11 workplace  ISPF Object/Action workplace

Enter X to Terminate using log/list defaults

F1=Help      F2=Split      F3=Exit      F6=TSO      F7=Backward  F8=Forward
F9=Swap      F10=Actions   F12=Recall
```

Figure 2: A 3270 Mainframe Screen

So, congratulations! You are now the owner of a huge company (complete with private jet and corporate limousine) that for the past 30 years or so has been developing programs that run on the Mainframe. You’ve spent huge – no, *enormous* amounts of money, time and effort to get them working for you. And after 30 years of debugging, tweaking, changing and modifying, your Mainframe programs are working fine. But more importantly, your business for the past 30 years has been evolving around these programs. So, the Mainframe has slowly become the core of your entire business. No Mainframe, no business. Welcome to the Mainframe world!

That's what most Mainframe users are facing. They've got these core systems running on Mainframes (like those managing your bank accounts), and they desperately, desperately need them.

If Mainframes Are So Good, Why Does Everyone Want to Get Rid of Them?

There must be hundreds of computing journalists today that see the IBM Z Mainframe as a computing Rasputin. Shoot him, poison him, drown him – and somehow, he keeps breathing! For many years computer magazines have delighted in writing about the death of IBM Z Mainframes - and many people believed them (and still do). But why don't people like them? And why aren't there many new Mainframe users? Simple: money, people, the world.

Money

Mainframes are expensive - very expensive. And we're not just talking about buying the Mainframe hardware and software. You're also up for:

- Hardware support costs – if they break, you need someone to come and fix them.
- Software licensing costs – the right to run the software, and support in case there's a problem with it.
- People to look after and maintain the Mainframe.
- A computer room – you can't put a Mainframe under your desk. They need an air-conditioned, climate-controlled computer room.

IBM has worked hard in recent years to reduce the costs of Mainframes and they've been quite successful. But the fact remains that Mainframes still aren't cheap.

People

Mainframes are far harder to look after than other computing options. Let's look at this for a second:

- You need more people to look after a Mainframe.
- Mainframes are harder to use, so these people need to have more training. A typical University graduate can administer a UNIX server without a great deal of training. The same graduate needs a couple of years training to become a competent Mainframe administrator.
- You can't find people who are Mainframe-savvy. Quality people with Mainframe skills or skills in programming older languages like COBOL are becoming harder to find. Few university graduates become skilled on the Mainframe, and existing Mainframe experts are retiring. We talk more about this later.

The World

Picture this: you've got all this data and lots of programs that access, process and change it. But Mainframes aren't pretty, and the world wants pretty applications. Applications that let people access your data from the internet. Applications that can make it look nice: drop down boxes, fancy graphics - the works.

Pretty isn't the only issue. Mainframes have a history of not playing well with other systems. In past years if you wanted to connect your Mainframe to other computers in the world or the internet, it was hard.

IBM has made a lot of changes to the Mainframe to fix this. For example, z/OS includes UNIX. That's right - a full, complete UNIX operating system is inside z/OS. This certainly means that you can run UNIX applications under z/OS. But perhaps more importantly, this gives you a path between your older Mainframe applications and UNIX oriented programs and equipment, including the internet. Nonetheless, connecting your Mainframe applications with newer systems and technology still isn't a straightforward task.

So, that sounds pretty convincing, doesn't it? Bring over that dump truck – we're getting rid of all our Mainframes! But before we start looking for extra-large garbage bags, let's have a look at the other side.

IBM as Snow White? The Mainframe Market Today

If you were in the market for a computer in the 1950s and 1960s, you were looking at Mainframes - there was really nothing else. And the dominant player was IBM.

The Mainframe market in those days was sometimes called 'IBM and the Seven Dwarfs' - the 'dwarfs' being the other major Mainframe manufacturers: Burroughs, Control Data, General Electric, Honeywell, NCR, RCA, and UNIVAC.

In some ways the Mainframe market today hasn't changed much - a dominant IBM together with manufacturers such as:

- Fujitsu (formerly Siemens) BS2000 running OSD.
- HPE NonStop server (originally Tandem NonStop).
- Unisys ClearPath running MCP (originally Burroughs) and OS220 (originally Univac-Sperry).
- Bull NovaScale 9000 Series running GCOS 8.

All these systems are still in use today and are fully supported by their vendors. Like the IBM Z they are legacy systems (meaning they're old technology but still needed), and most have been developed so they can also run UNIX, and in some cases even Microsoft Windows.

Users of these Mainframes are faced with many of the same issues as IBM Z Mainframe users – high costs, web-enablement, future development, future support, and finding skilled technical people.

Why Keep Mainframes?

Here's an interesting fact - the amount of work Mainframes are doing around the world is *increasing*. That's right, though the number of Mainframes being used has dropped over the past years, if you add up all the processing they do – it's increasing. In fact, you will find that most major banks, insurance companies, and larger government departments in more industrialized countries have a Mainframe or two hidden in the computer room closet. So why haven't they moved on?

Two reasons: it's hard to move, and Mainframes do some things better.

The Y2K 'Bug' – A Mainframe Disease

In the 1990s, the Y2K 'bug' made worldwide headlines. Many were predicting catastrophic problems on January 1, 2000, for older applications – especially those running on Mainframes.

This was caused by many older applications using just two digits to store the year. For example, 1999 was stored as '99'. So, in 2000, some applications would think that it was 1900.

Many organizations were facing the huge cost of changing their older Mainframe applications – and decided not to. Instead, they spent their money on moving their applications away from the Mainframe.

It's Hard to Move Away From Mainframes

Remember? You're the owner of that organization that first invested in Mainframes 30 years ago? All the Mainframe programs and systems have been evolving to suit your business. And what's more, your business has been evolving around the Mainframe. You're joined at the hip. The Mainframe and its applications are the core of your business – holding the critical information. You simply cannot do without the functionality or the data.

To move away from the Mainframe, you must:

- Buy new hardware – and this hardware needs to be as reliable as the Mainframe hardware.
- Buy new systems software – and this needs to be reliable too.
- Find people who can administer your new systems.
- Create new procedures to administer your new systems. For example: backups, database maintenance, security, audit, and Disaster Recovery.
- Either rewrite all your programs (that you've spent 30 years developing and modifying) on another platform and in another language or modify an 'off the shelf package' - software already written. And it all must work.

- Train all your users to use the new system.
- Migrate all your data from the Mainframe to the newer system.
- Find something to do with all the Mainframe people who are now out of a job.

Also remember that you need to cutover from your old system to a new one. You're going to have to shut down the Mainframe systems, migrate your existing data across, then startup the new system – and hope it works.

Oh yes, we almost forgot about cost – none of this is going to be cheap. So, you're going to need a very good reason to make this change.

We talk more about replacing Mainframes in Chapter 9: What Do We Do With It?

Why Are Mainframes Using Older Technology?

So why are Mainframes using old technology - why can't they use the new stuff? Mainframes actually use state-of-the-art technology, but still must support the old stuff because of *Backward Compatibility*.

In 1964, IBM released a new Mainframe - the System/360. But there was one problem - the System/360 wouldn't run programs written on the previous IBM machines: customers had to rewrite them. This isn't great.

Today, every time a new Mainframe is released it must work with existing Mainframe application programs and software - it has to be *backward compatible*. This makes it more difficult for IBM to move to newer technology.

Mainframes Do Some Things Better

Think about ambulances. If you're running an ambulance service, are you going to buy cheap or used vehicles to use as ambulances? No way! You're going to want to spend money on vehicles that you know won't break down. And what's more, you're going to want to spend lots of money looking after and maintaining those vehicles because if they fail, people die.

In many cases, the Mainframe is like this (except that no-one dies). Yes, it's a system based on older technology. But in some ways, it is the best at what it does. Its strengths are:

- **Reliability:** Would you believe that the 'z' in IBM Z stands for 'zero downtime?' Although this is a bit of marketing, there's no doubt that Mainframes run, and run, and run.

Mainframe hardware is the most reliable on the planet – and the best supported. If it breaks, you'll have an engineer at your doorstep in a flash, 24 hours a day. Mainframe software has been used for the critical business applications of many large corporations for a long, long time. Today, an organization with a properly setup and run Mainframe can measure the time between unscheduled outages in YEARS.

- **Accountability:** The z/OS operating system has the best recording of any operating system on the planet. This recording has little overhead, and lets you monitor everything from performance (how fast did we do that?), to security (who tried to do that?) and accounting (who and how much do we charge for that?).
- **Data integrity:** Yes, we're back to this. When data integrity is important you can't beat the Mainframe. The database applications that run on z/OS such as IMS and Db2 have lots of features to make sure that database corruption simply doesn't occur. So, a database on the Mainframe is very, very safe.
- **Throughput:** Even today, it's difficult to find another platform and database management system that can process the high number of transactions that Mainframe database applications eat up for breakfast. We're talking thousands of transactions PER SECOND, each finishing in less than half a second. So, for larger applications the Mainframe still seems like home.
- **Security:** Correctly setup Mainframes are the most secure computing system on the planet. Period. No viruses on the Mainframe.

The Last Word

So, we now know about Mainframes and where they sit in the computing industry. We know that:

- They're usually legacy systems holding the key data for an organization.
- They can do a lot of work, and they can do it quickly, safely and reliably.
- They're expensive, and difficult to set up and run.
- They do some things far better than any other system.
- They're hard to move from.

The fact is that the IBM Z Mainframe is going to be around for many, many years to come.