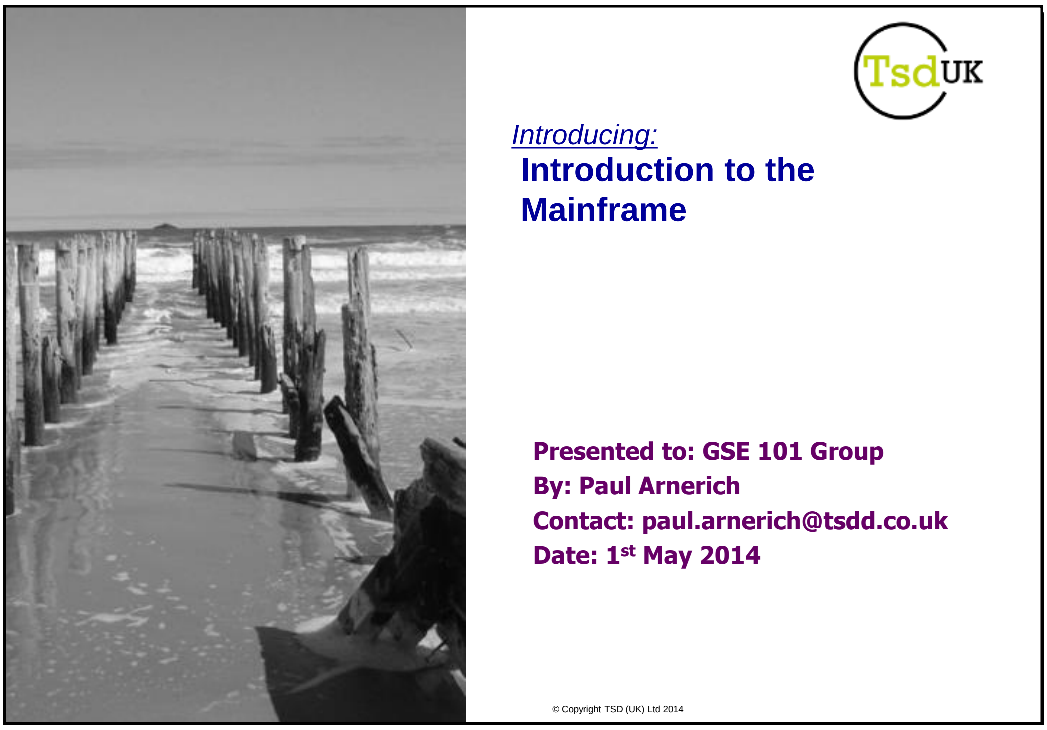
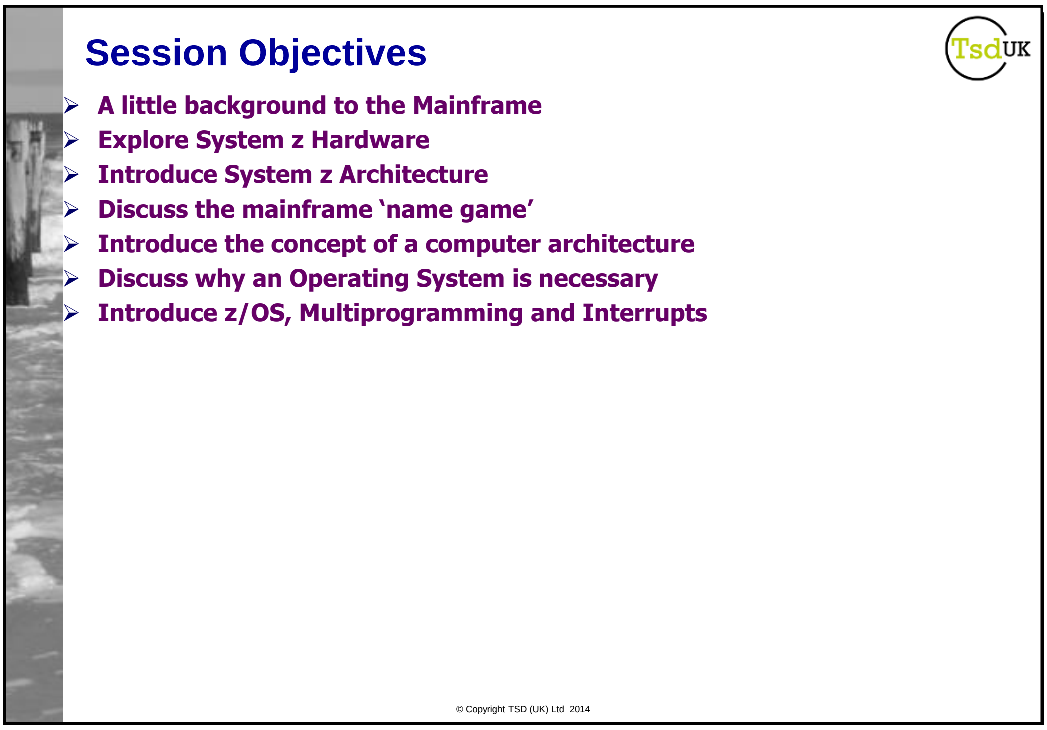


Introducing:

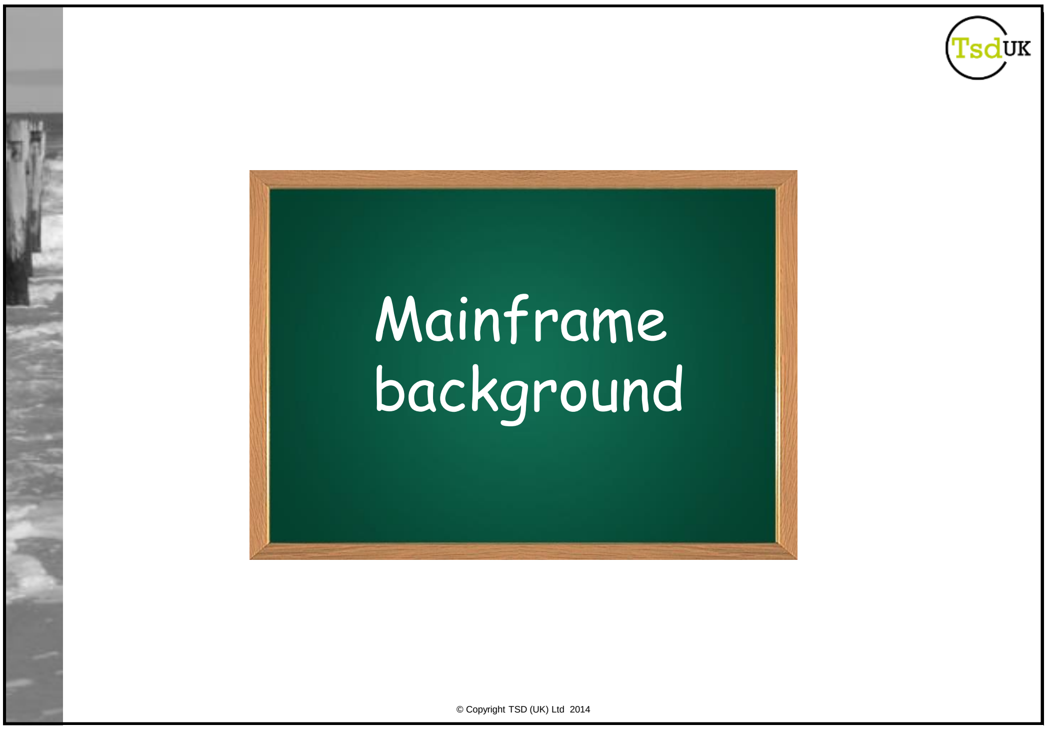
Introduction to the Mainframe

Presented to: GSE 101 Group
By: Paul Arnerich
Contact: paul.arnerich@tsdd.co.uk
Date: 1st May 2014

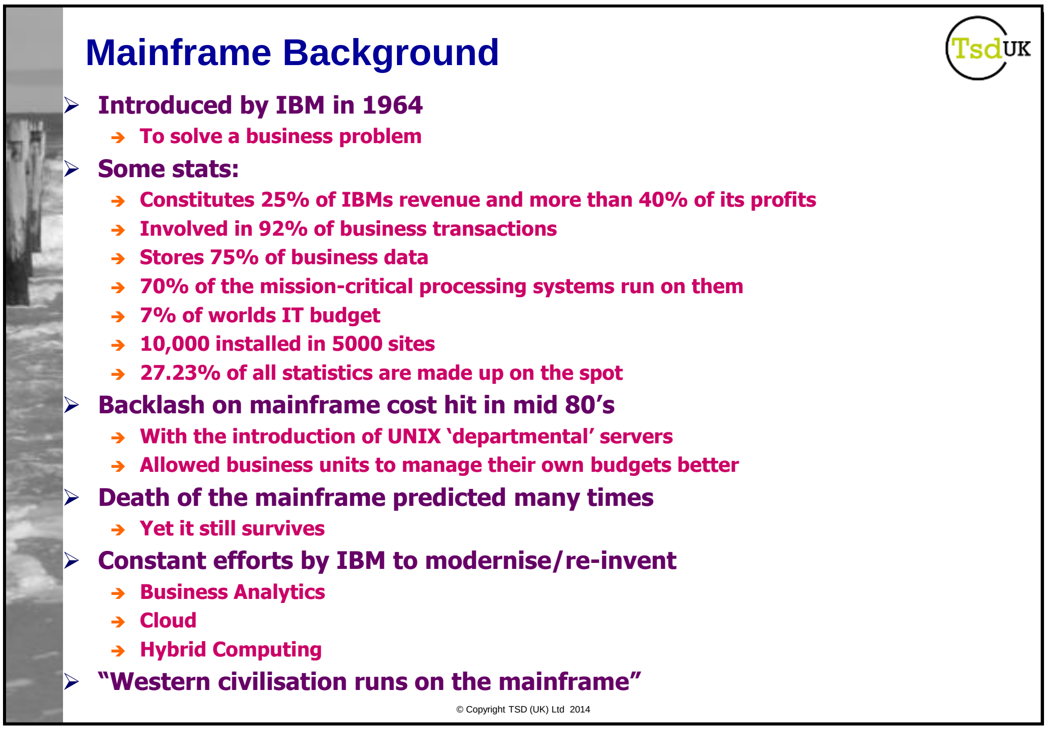


Session Objectives

- A little background to the Mainframe
- Explore System z Hardware
- Introduce System z Architecture
- Discuss the mainframe ‘name game’
- Introduce the concept of a computer architecture
- Discuss why an Operating System is necessary
- Introduce z/OS, Multiprogramming and Interrupts

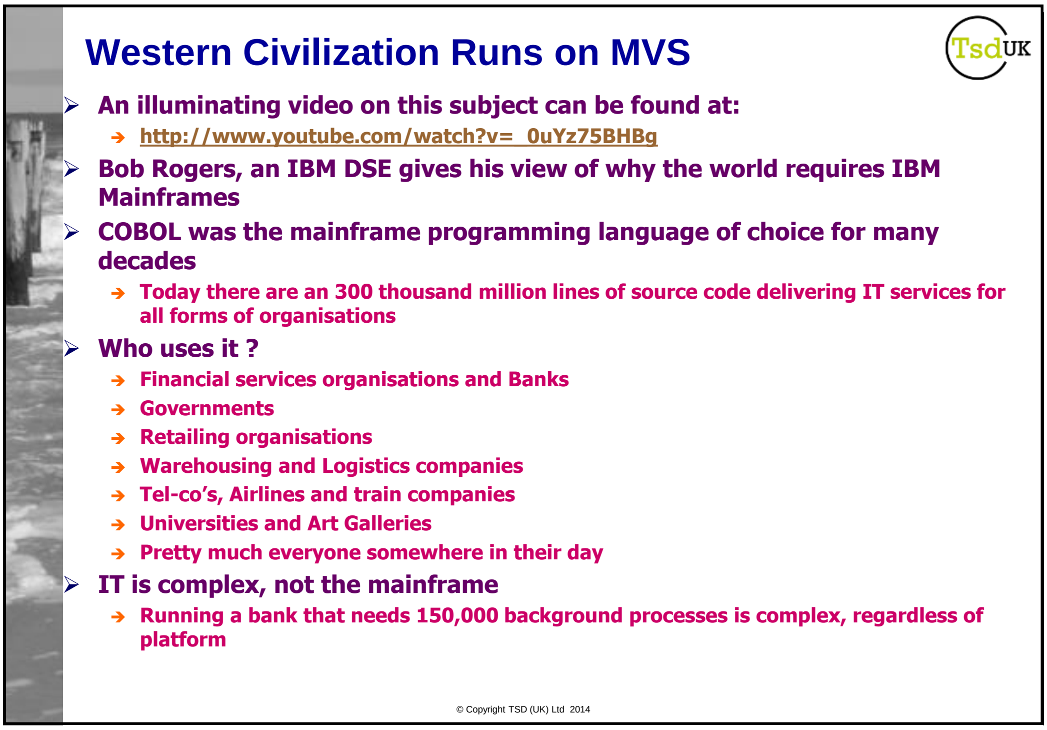


Mainframe background



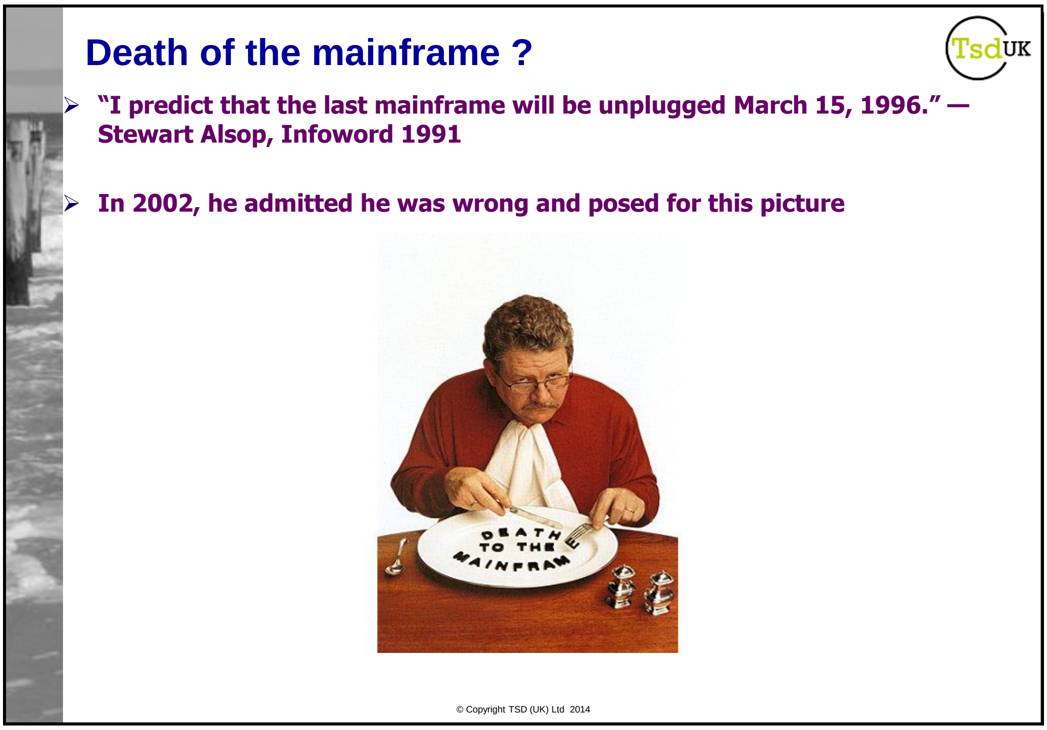
Mainframe Background

- Introduced by IBM in 1964
  - To solve a business problem
- Some stats:
  - Constitutes 25% of IBMs revenue and more than 40% of its profits
  - Involved in 92% of business transactions
  - Stores 75% of business data
  - 70% of the mission-critical processing systems run on them
  - 7% of worlds IT budget
  - 10,000 installed in 5000 sites
  - 27.23% of all statistics are made up on the spot
- Backlash on mainframe cost hit in mid 80's
  - With the introduction of UNIX 'departmental' servers
  - Allowed business units to manage their own budgets better
- Death of the mainframe predicted many times
  - Yet it still survives
- Constant efforts by IBM to modernise/re-invent
  - Business Analytics
  - Cloud
  - Hybrid Computing
- "Western civilisation runs on the mainframe"



Western Civilization Runs on MVS

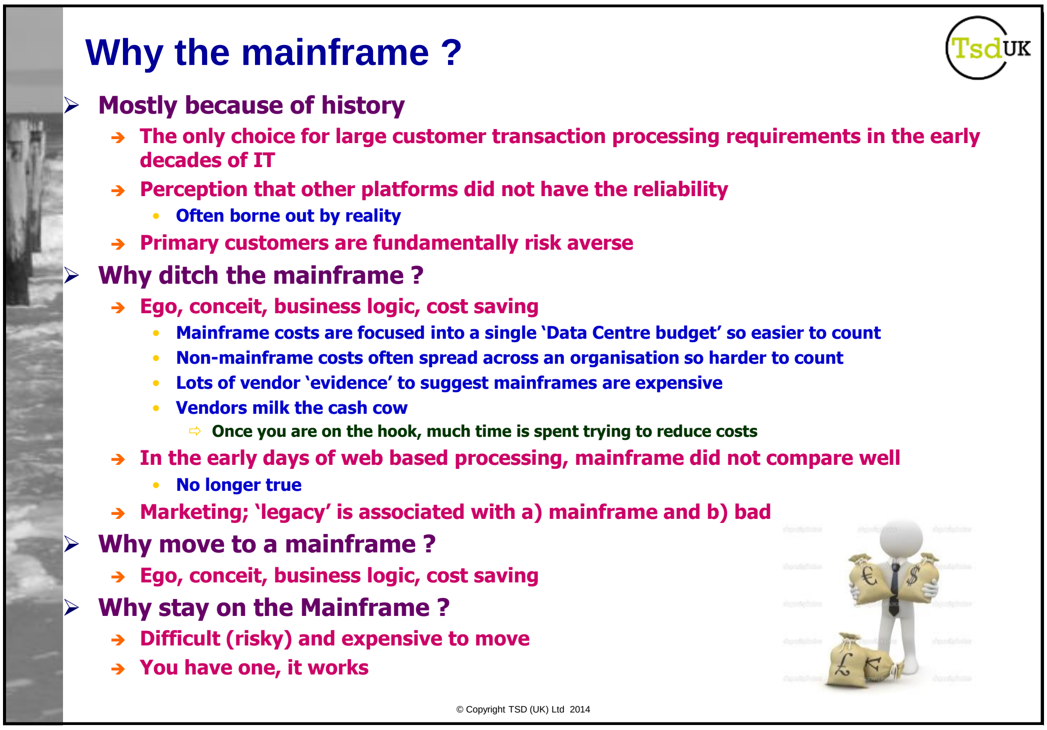
- An illuminating video on this subject can be found at:
  - http://www.youtube.com/watch?v=_0uYz75BHBg
- Bob Rogers, an IBM DSE gives his view of why the world requires IBM Mainframes
- COBOL was the mainframe programming language of choice for many decades
  - Today there are an 300 thousand million lines of source code delivering IT services for all forms of organisations
- Who uses it ?
  - Financial services organisations and Banks
  - Governments
  - Retailing organisations
  - Warehousing and Logistics companies
  - Tel-co's, Airlines and train companies
  - Universities and Art Galleries
  - Pretty much everyone somewhere in their day
- IT is complex, not the mainframe
  - Running a bank that needs 150,000 background processes is complex, regardless of platform



Death of the mainframe ?

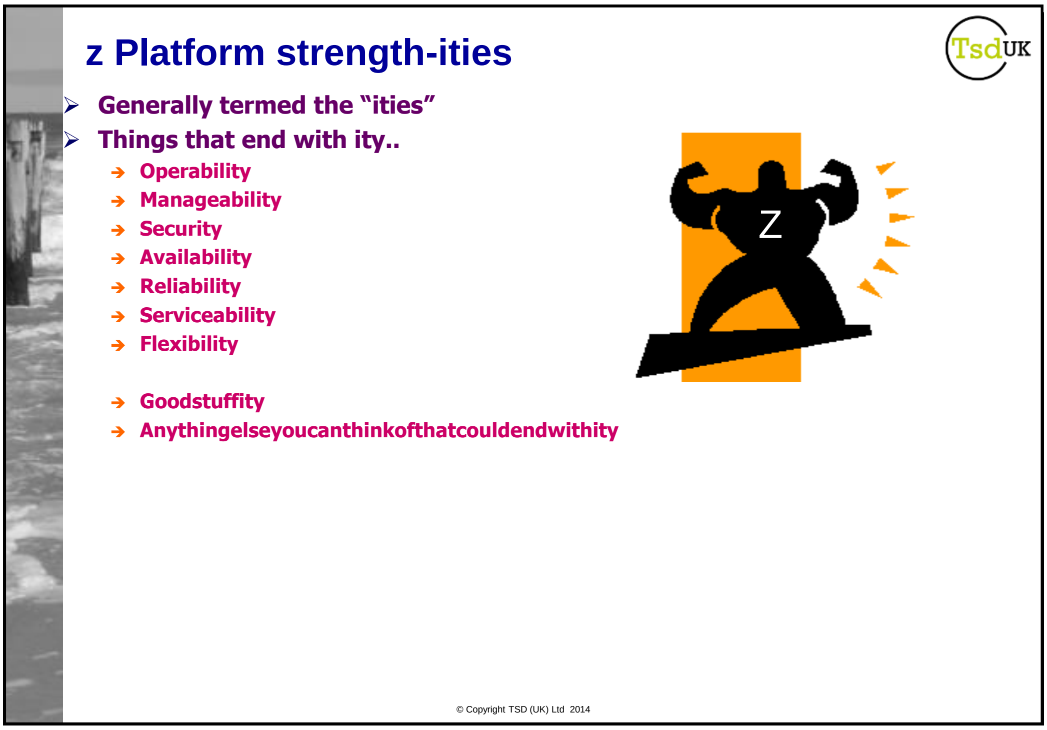
- "I predict that the last mainframe will be unplugged March 15, 1996." — Stewart Alsop, Infoword 1991
- In 2002, he admitted he was wrong and posed for this picture

DEATH
TO THE
MAINFRAME



Why the mainframe ?

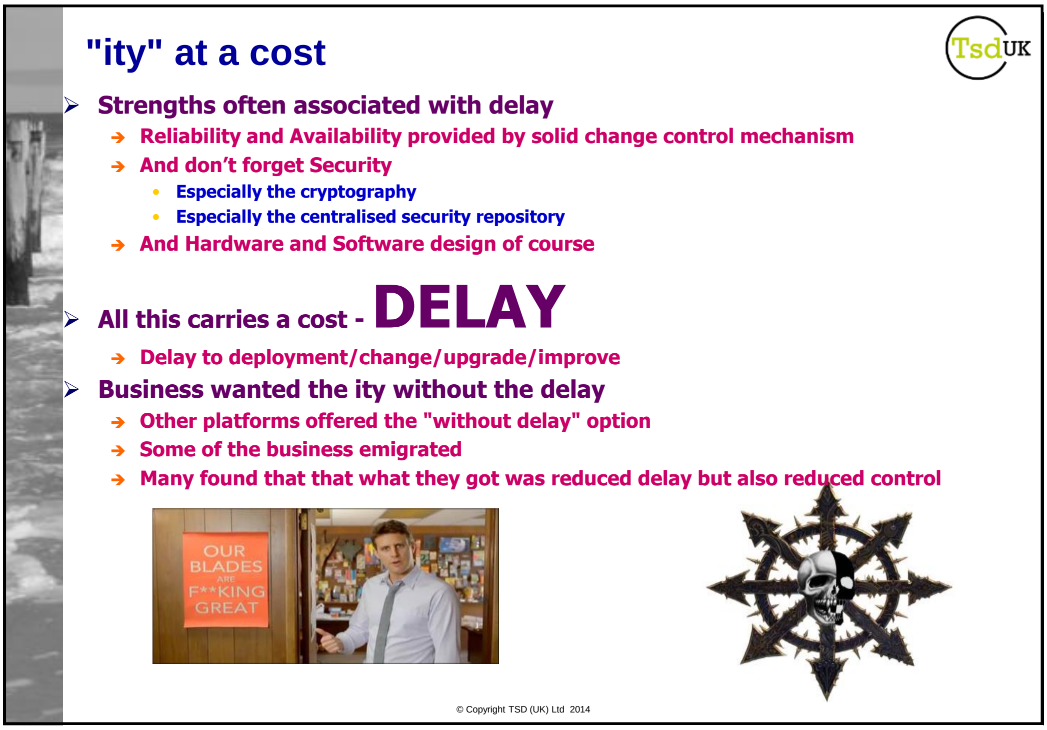
- Mostly because of history
  - The only choice for large customer transaction processing requirements in the early decades of IT
  - Perception that other platforms did not have the reliability
    - Often borne out by reality
  - Primary customers are fundamentally risk averse
- Why ditch the mainframe ?
  - Ego, conceit, business logic, cost saving
    - Mainframe costs are focused into a single 'Data Centre budget' so easier to count
    - Non-mainframe costs often spread across an organisation so harder to count
    - Lots of vendor 'evidence' to suggest mainframes are expensive
    - Vendors milk the cash cow
      - Once you are on the hook, much time is spent trying to reduce costs
  - In the early days of web based processing, mainframe did not compare well
    - No longer true
  - Marketing; 'legacy' is associated with a) mainframe and b) bad
- Why move to a mainframe ?
  - Ego, conceit, business logic, cost saving
- Why stay on the Mainframe ?
  - Difficult (risky) and expensive to move
  - You have one, it works



z Platform strength-ities

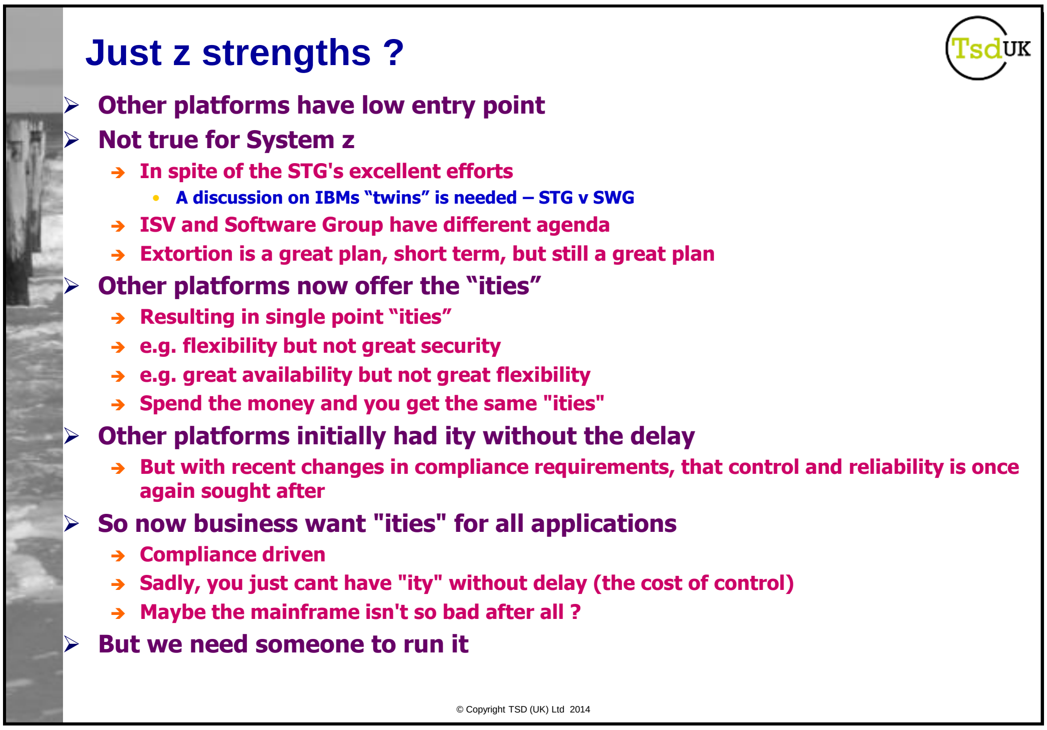
- Generally termed the "ities"
- Things that end with ity..
  - Operability
  - Manageability
  - Security
  - Availability
  - Reliability
  - Serviceability
  - Flexibility

  - Goodstuffity
  - Anythingelseyoucanthinkofthatcouldendwithity



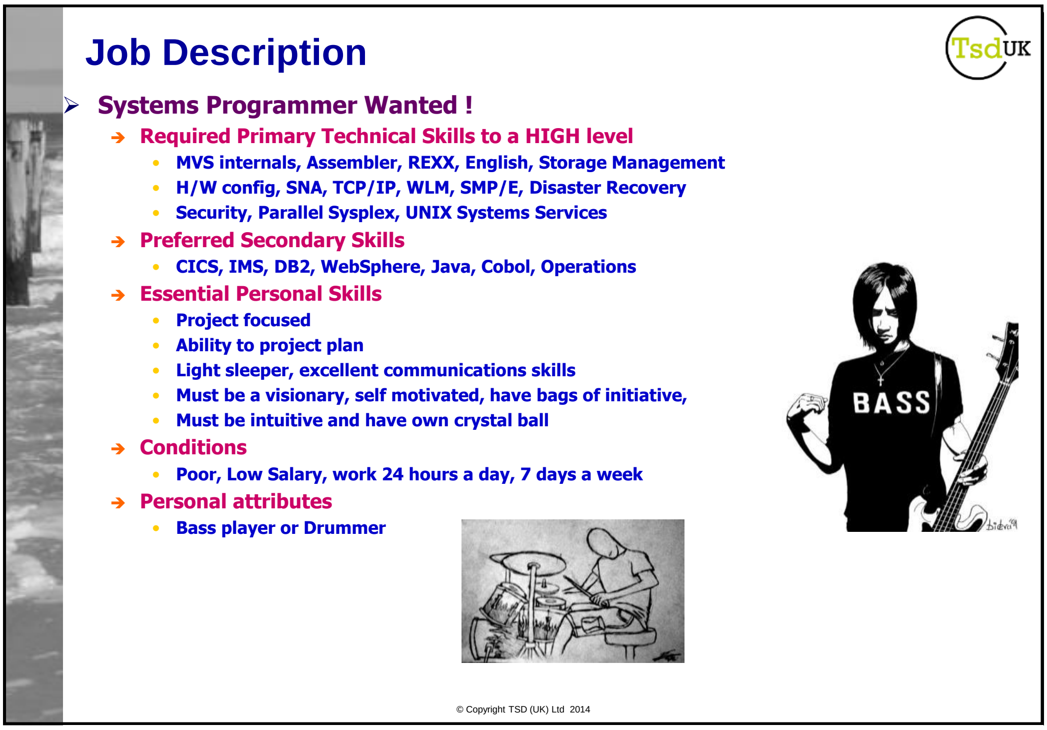
"ity" at a cost

- Strengths often associated with delay
  - Reliability and Availability provided by solid change control mechanism
  - And don't forget Security
    - Especially the cryptography
    - Especially the centralised security repository
  - And Hardware and Software design of course
- All this carries a cost - **DELAY**
  - Delay to deployment/change/upgrade/improve
- Business wanted the ity without the delay
  - Other platforms offered the "without delay" option
  - Some of the business emigrated
  - Many found that that what they got was reduced delay but also reduced control



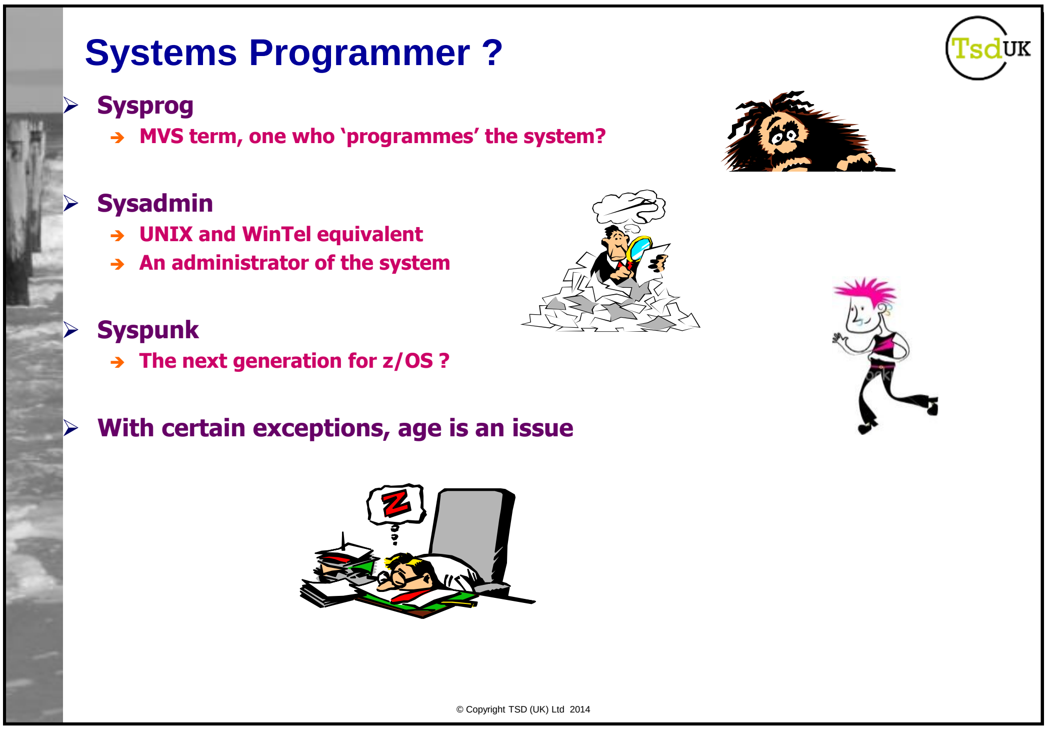
Just z strengths ?

- Other platforms have low entry point
- Not true for System z
  - In spite of the STG's excellent efforts
    - A discussion on IBMs "twins" is needed – STG v SWG
  - ISV and Software Group have different agenda
  - Extortion is a great plan, short term, but still a great plan
- Other platforms now offer the "ities"
  - Resulting in single point "ities"
  - e.g. flexibility but not great security
  - e.g. great availability but not great flexibility
  - Spend the money and you get the same "ities"
- Other platforms initially had ity without the delay
  - But with recent changes in compliance requirements, that control and reliability is once again sought after
- So now business want "ities" for all applications
  - Compliance driven
  - Sadly, you just cant have "ity" without delay (the cost of control)
  - Maybe the mainframe isn't so bad after all ?
- But we need someone to run it



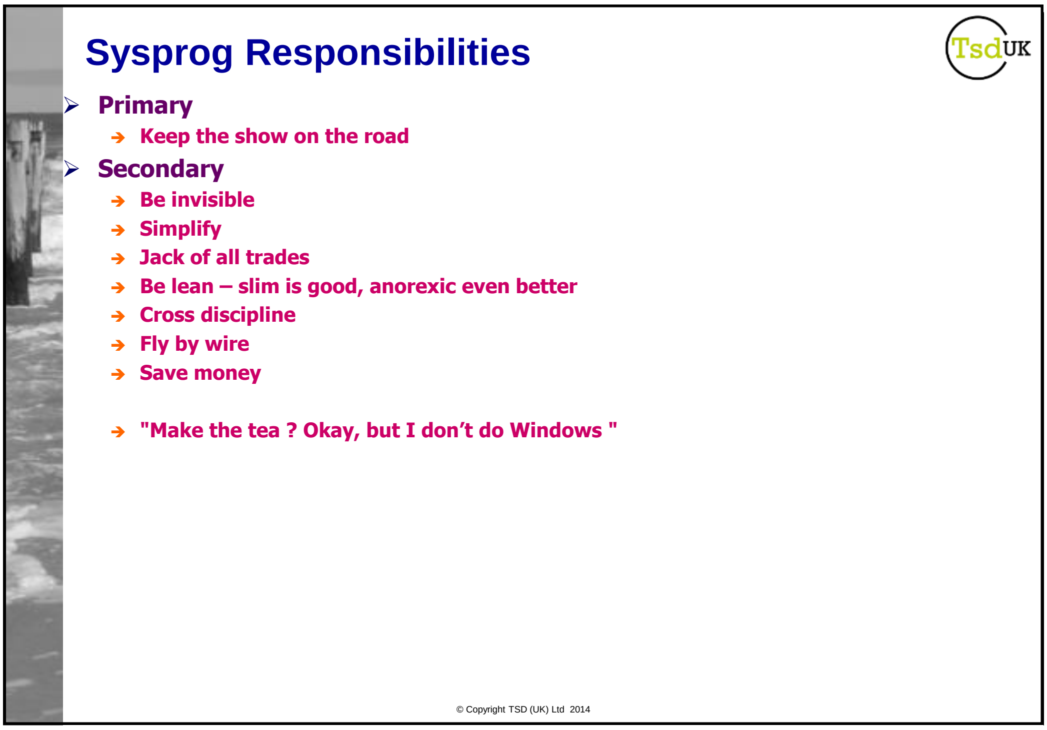
Job Description

- **Systems Programmer Wanted !**
  - **Required Primary Technical Skills to a HIGH level**
    - **MVS internals, Assembler, REXX, English, Storage Management**
    - **H/W config, SNA, TCP/IP, WLM, SMP/E, Disaster Recovery**
    - **Security, Parallel Sysplex, UNIX Systems Services**
  - **Preferred Secondary Skills**
    - **CICS, IMS, DB2, WebSphere, Java, Cobol, Operations**
  - **Essential Personal Skills**
    - **Project focused**
    - **Ability to project plan**
    - **Light sleeper, excellent communications skills**
    - **Must be a visionary, self motivated, have bags of initiative,**
    - **Must be intuitive and have own crystal ball**
  - **Conditions**
    - **Poor, Low Salary, work 24 hours a day, 7 days a week**
  - **Personal attributes**
    - **Bass player or Drummer**



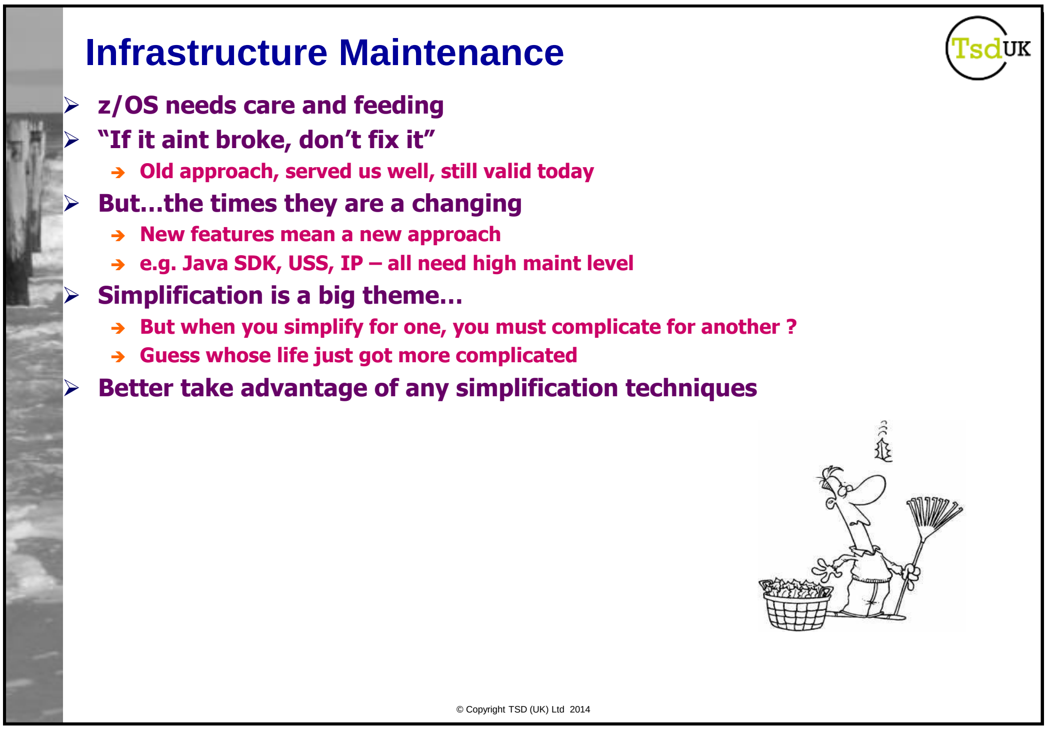
Systems Programmer ?

- **Sysprog**
  - **MVS term, one who 'programmes' the system?**
- **Sysadmin**
  - **UNIX and WinTel equivalent**
  - **An administrator of the system**
- **Syspunk**
  - **The next generation for z/OS ?**
- **With certain exceptions, age is an issue**



Sysprog Responsibilities

- Primary
  - Keep the show on the road
- Secondary
  - Be invisible
  - Simplify
  - Jack of all trades
  - Be lean – slim is good, anorexic even better
  - Cross discipline
  - Fly by wire
  - Save money
  - "Make the tea ? Okay, but I don't do Windows "

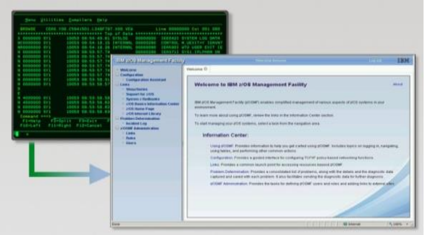


Infrastructure Maintenance

- z/OS needs care and feeding
- "If it aint broke, don't fix it"
  - Old approach, served us well, still valid today
- But…the times they are a changing
  - New features mean a new approach
  - e.g. Java SDK, USS, IP – all need high maint level
- Simplification is a big theme…
  - But when you simplify for one, you must complicate for another ?
  - Guess whose life just got more complicated
- Better take advantage of any simplification techniques

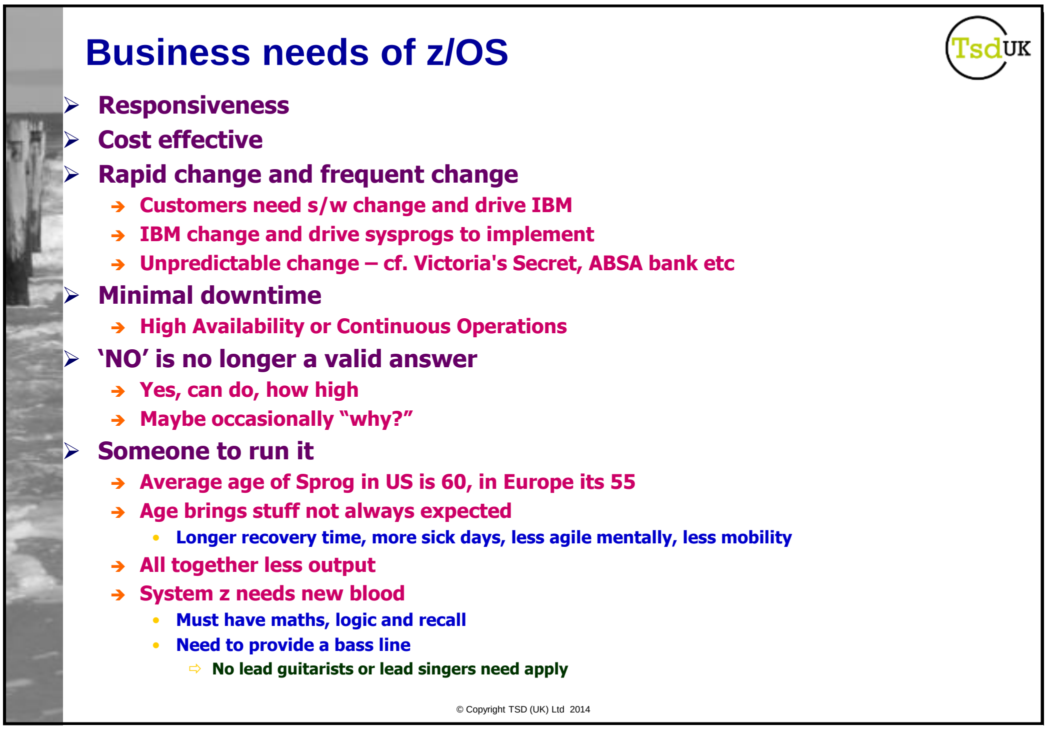
Sys Prog or Admin ?

- Rare to spend time programming today
  - The odd exit, limited customer BALR code
- IBM and management want admin not prog
- Simplification of administration is a big theme
  - CSI break-up – never delivered
  - MSYS for Setup – withdrawn after 4 years
  - Wizards - exist but lightly used
  - Java client based install and config - growing
    - Not very successful so far, but..
- Project eLiza
  - Autonomic computing
  - Self healing, self optimising, self managing
  - Shameless Selfless, self aggrandisement ?
  - z/OS Management Facility (zOSMF)
    - Target audience is India and China



Philosophical shift

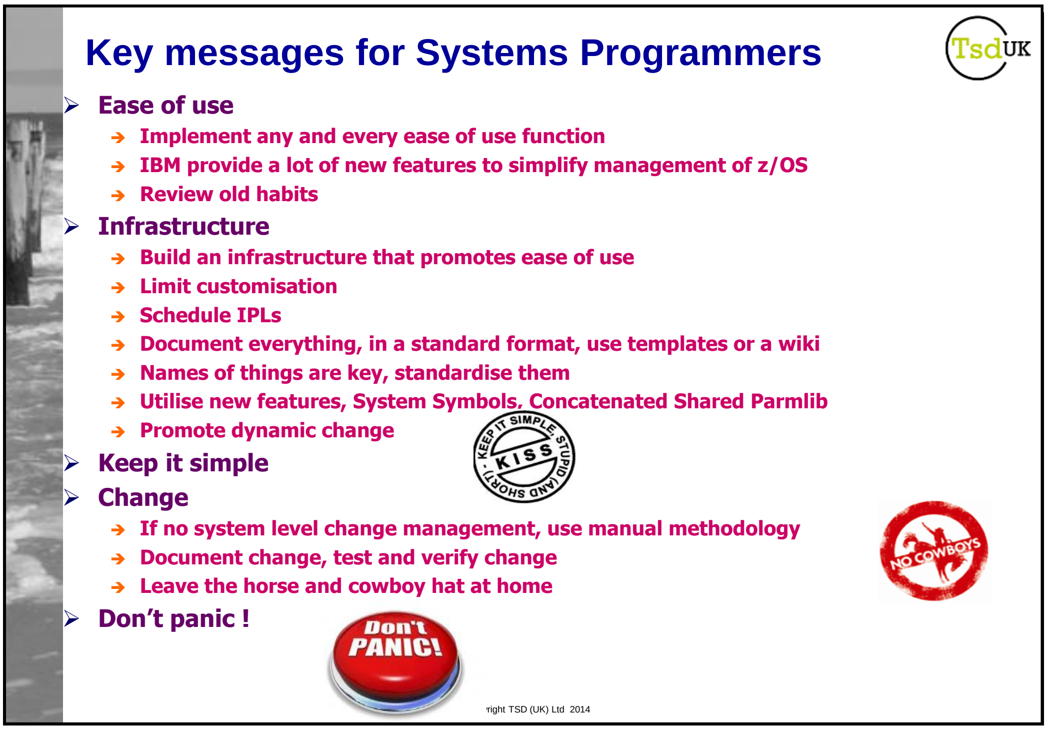
- 70’s & 80’s
  - Lots of unintentional job preservations
    - “What do I do ? If I tell you, I have to kill you”
  - Too complex to disseminate
  - King of the roost
  - Attitude prevalent that the business exists to support IT
- 90’s, 00’s & 10’s
  - Leaner, pressure from other platforms
  - Far greater knowledge of IT in the wild
    - “Oh yes, I have a computer at home”
  - Attitude is “IT exists to serve the business”
    - Notice it's no longer 'supporting', but 'serving' ?
  - More business focused
  - Implementation activities based on business needs



Business needs of z/OS

- Responsiveness
- Cost effective
- Rapid change and frequent change
  - Customers need s/w change and drive IBM
  - IBM change and drive sysprogs to implement
  - Unpredictable change – cf. Victoria's Secret, ABSA bank etc
- Minimal downtime
  - High Availability or Continuous Operations
- ‘NO’ is no longer a valid answer
  - Yes, can do, how high
  - Maybe occasionally “why?”
- Someone to run it
  - Average age of Sprog in US is 60, in Europe its 55
  - Age brings stuff not always expected
    - Longer recovery time, more sick days, less agile mentally, less mobility
  - All together less output
  - System z needs new blood
    - Must have maths, logic and recall
    - Need to provide a bass line
      - No lead guitarists or lead singers need apply

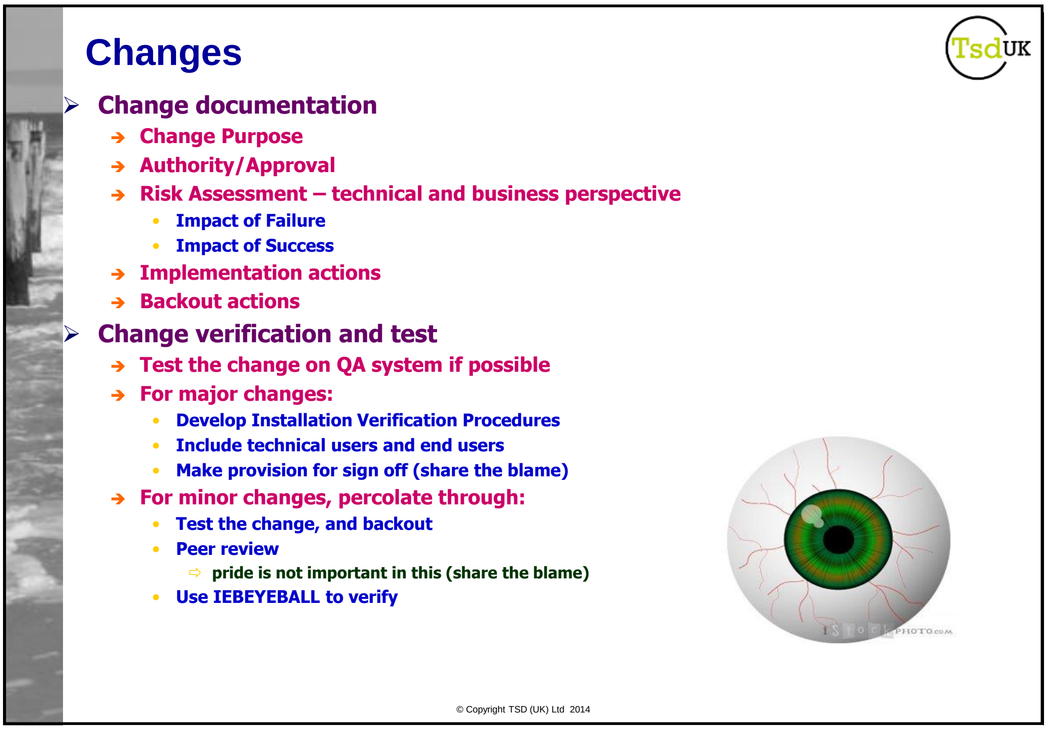
© Copyright TSD (UK) Ltd 2014



Key messages for Systems Programmers

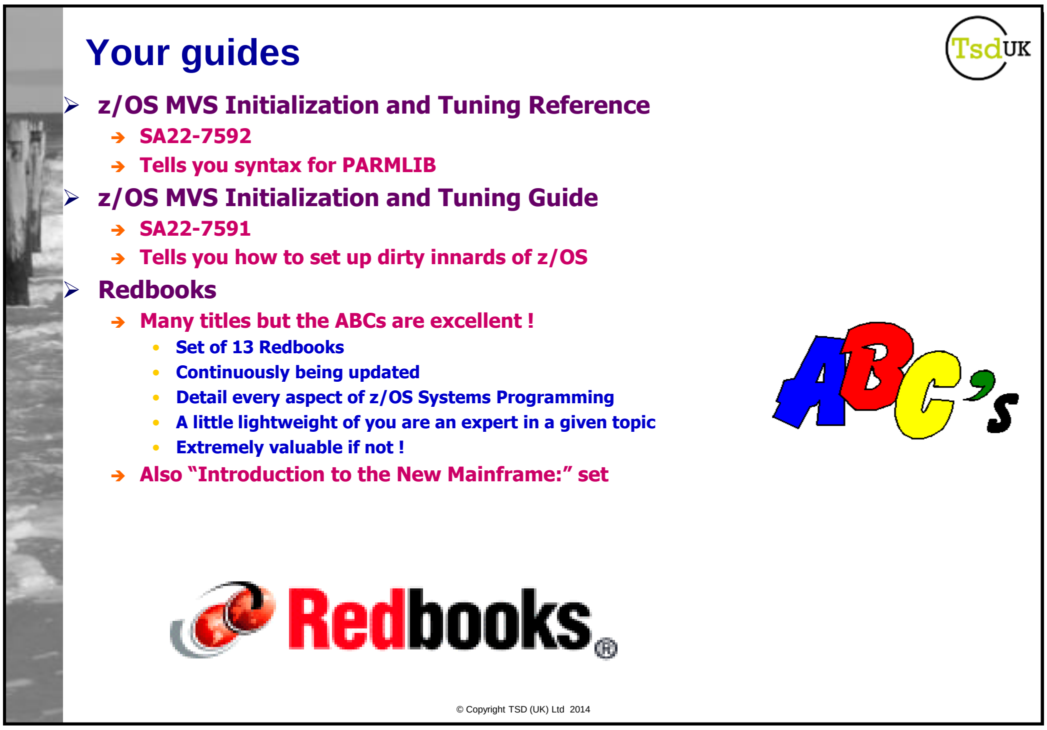
- Ease of use
  - Implement any and every ease of use function
  - IBM provide a lot of new features to simplify management of z/OS
  - Review old habits
- Infrastructure
  - Build an infrastructure that promotes ease of use
  - Limit customisation
  - Schedule IPLs
  - Document everything, in a standard format, use templates or a wiki
  - Names of things are key, standardise them
  - Utilise new features, System Symbols, Concatenated Shared Parmlib
  - Promote dynamic change
- Keep it simple
- Change
  - If no system level change management, use manual methodology
  - Document change, test and verify change
  - Leave the horse and cowboy hat at home
- Don’t panic !

right TSD (UK) Ltd 2014



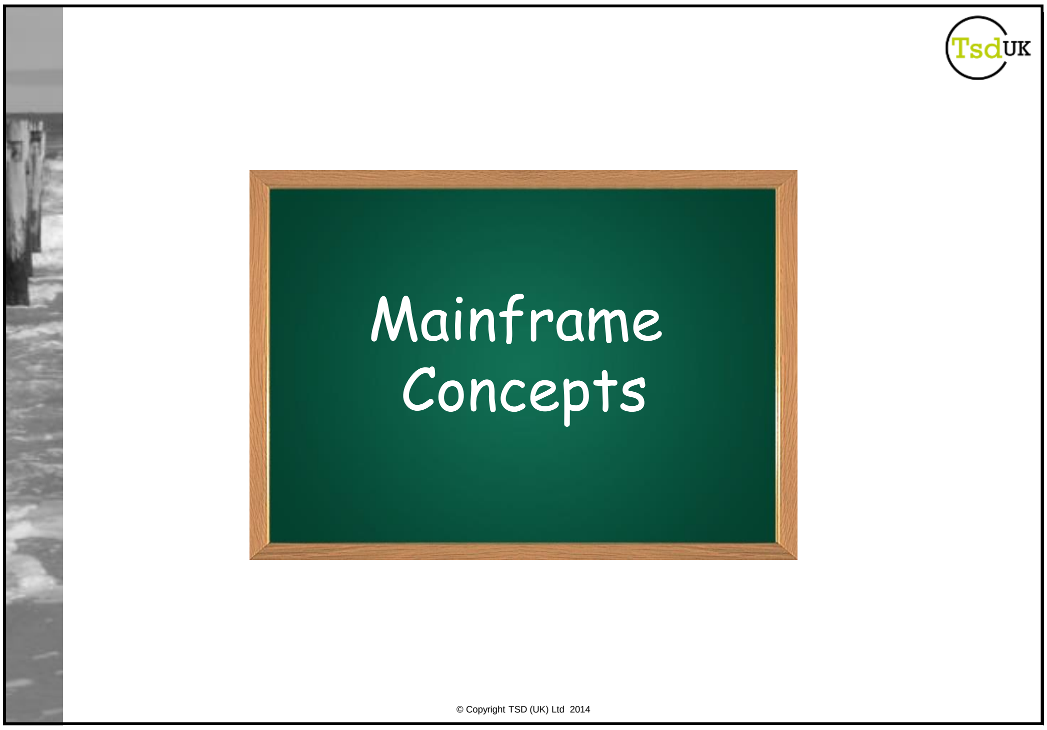
Changes

- Change documentation
  - Change Purpose
  - Authority/Approval
  - Risk Assessment – technical and business perspective
    - Impact of Failure
    - Impact of Success
  - Implementation actions
  - Backout actions
- Change verification and test
  - Test the change on QA system if possible
  - For major changes:
    - Develop Installation Verification Procedures
    - Include technical users and end users
    - Make provision for sign off (share the blame)
  - For minor changes, percolate through:
    - Test the change, and backout
    - Peer review
      - pride is not important in this (share the blame)
    - Use IEBEYEBALL to verify

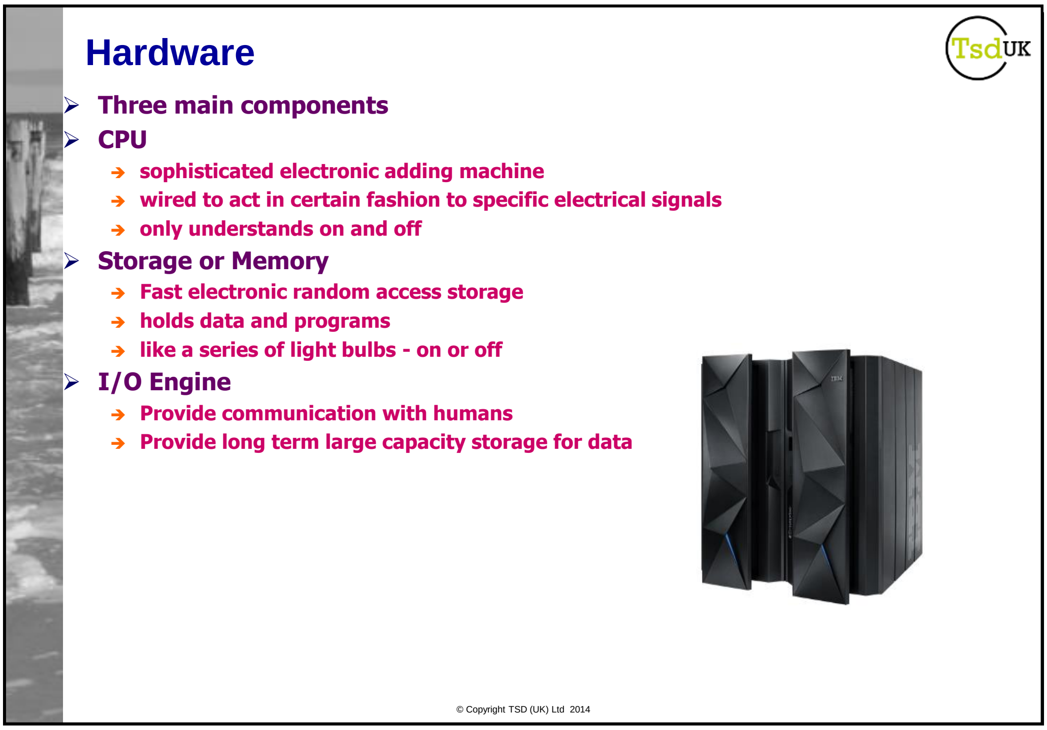


Your guides

- z/OS MVS Initialization and Tuning Reference
  - SA22-7592
  - Tells you syntax for PARMLIB
- z/OS MVS Initialization and Tuning Guide
  - SA22-7591
  - Tells you how to set up dirty innards of z/OS
- Redbooks
  - Many titles but the ABCs are excellent !
    - Set of 13 Redbooks
    - Continuously being updated
    - Detail every aspect of z/OS Systems Programming
    - A little lightweight of you are an expert in a given topic
    - Extremely valuable if not !
  - Also “Introduction to the New Mainframe:” set

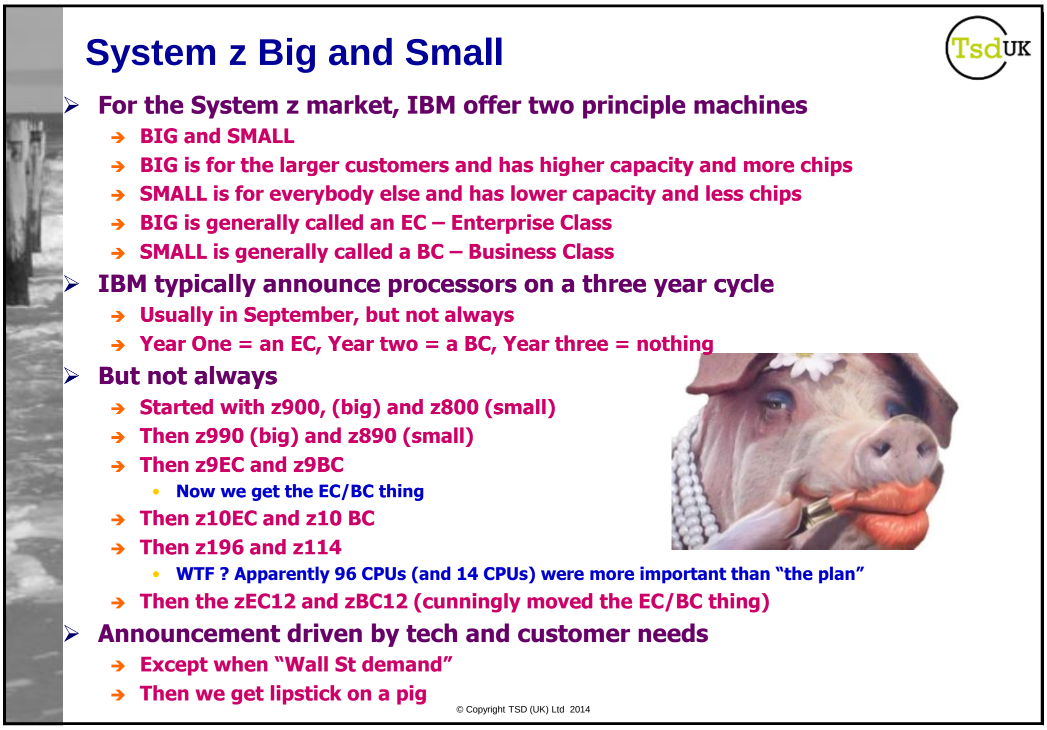


Mainframe Concepts



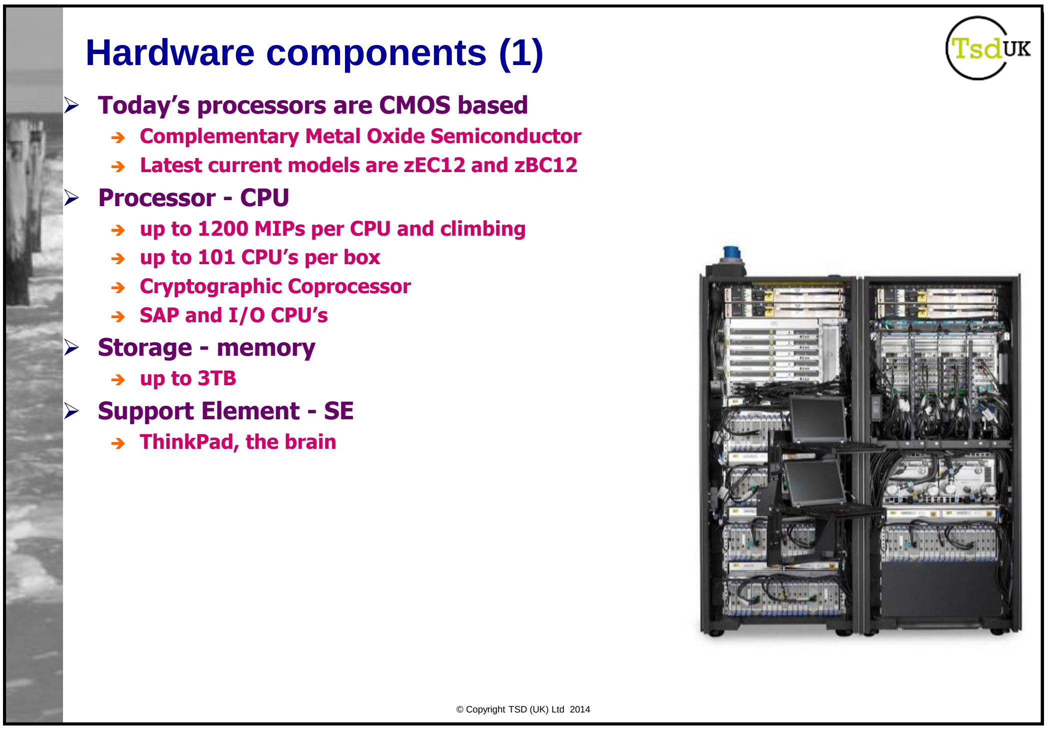
Hardware

- Three main components
- CPU
  - → sophisticated electronic adding machine
  - → wired to act in certain fashion to specific electrical signals
  - → only understands on and off
- Storage or Memory
  - → Fast electronic random access storage
  - → holds data and programs
  - → like a series of light bulbs - on or off
- I/O Engine
  - → Provide communication with humans
  - → Provide long term large capacity storage for data



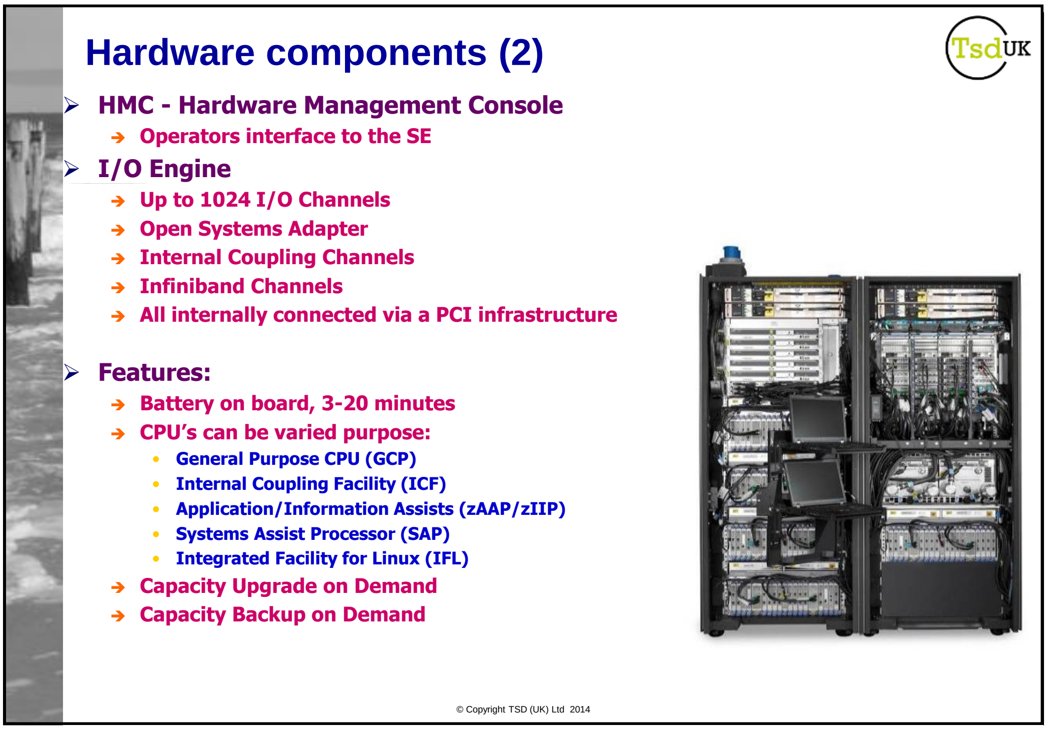
System z Big and Small

- For the System z market, IBM offer two principle machines
  - BIG and SMALL
  - BIG is for the larger customers and has higher capacity and more chips
  - SMALL is for everybody else and has lower capacity and less chips
  - BIG is generally called an EC – Enterprise Class
  - SMALL is generally called a BC – Business Class
- IBM typically announce processors on a three year cycle
  - Usually in September, but not always
  - Year One = an EC, Year two = a BC, Year three = nothing
- But not always
  - Started with z900, (big) and z800 (small)
  - Then z990 (big) and z890 (small)
  - Then z9EC and z9BC
    - Now we get the EC/BC thing
  - Then z10EC and z10 BC
  - Then z196 and z114
    - WTF ? Apparently 96 CPUs (and 14 CPUs) were more important than "the plan"
  - Then the zEC12 and zBC12 (cunningly moved the EC/BC thing)
- Announcement driven by tech and customer needs
  - Except when "Wall St demand"
  - Then we get lipstick on a pig



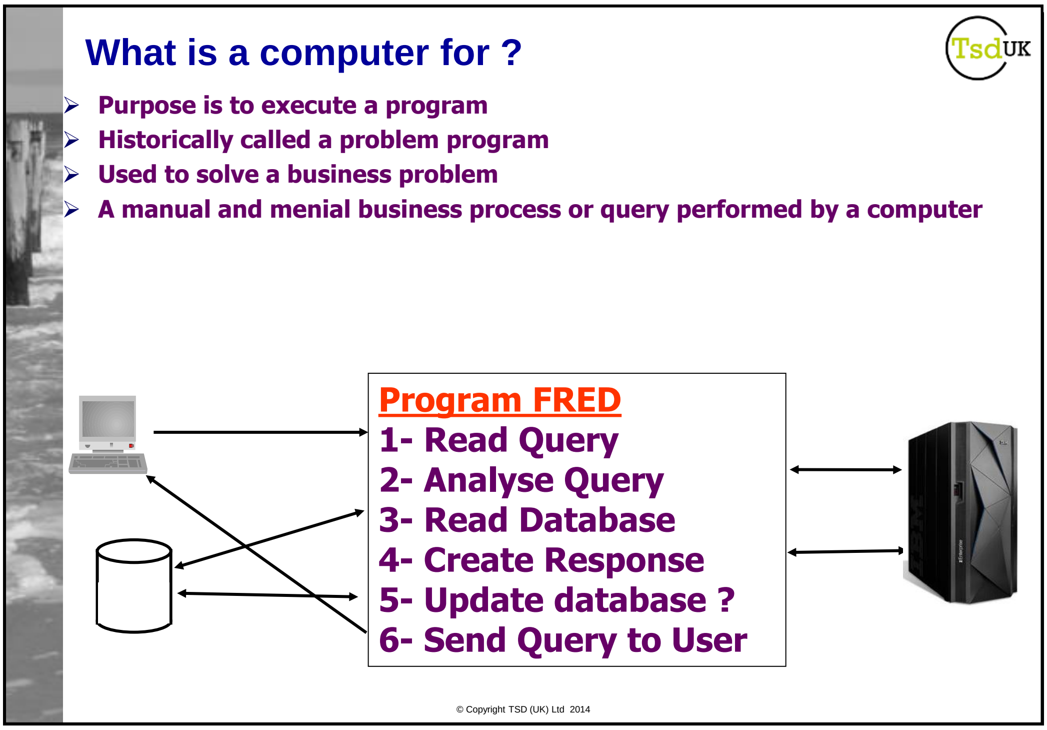
Hardware components (1)

- Today's processors are CMOS based
  - Complementary Metal Oxide Semiconductor
  - Latest current models are zEC12 and zBC12
- Processor - CPU
  - up to 1200 MIPs per CPU and climbing
  - up to 101 CPU's per box
  - Cryptographic Coprocessor
  - SAP and I/O CPU's
- Storage - memory
  - up to 3TB
- Support Element - SE
  - ThinkPad, the brain



Hardware components (2)

- HMC - Hardware Management Console
  - Operators interface to the SE
- I/O Engine
  - Up to 1024 I/O Channels
  - Open Systems Adapter
  - Internal Coupling Channels
  - Infiniband Channels
  - All internally connected via a PCI infrastructure
- Features:
  - Battery on board, 3-20 minutes
  - CPU's can be varied purpose:
    - General Purpose CPU (GCP)
    - Internal Coupling Facility (ICF)
    - Application/Information Assists (zAAP/zIIP)
    - Systems Assist Processor (SAP)
    - Integrated Facility for Linux (IFL)
  - Capacity Upgrade on Demand
  - Capacity Backup on Demand

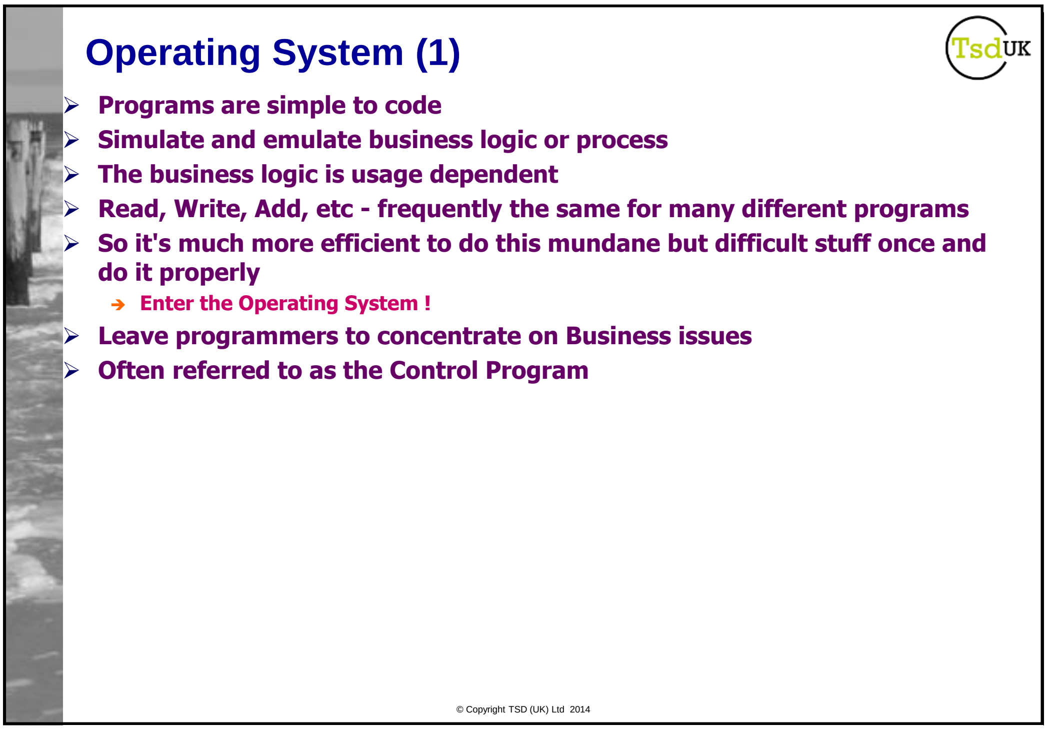


What is a computer for ?

- Purpose is to execute a program
- Historically called a problem program
- Used to solve a business problem
- A manual and menial business process or query performed by a computer

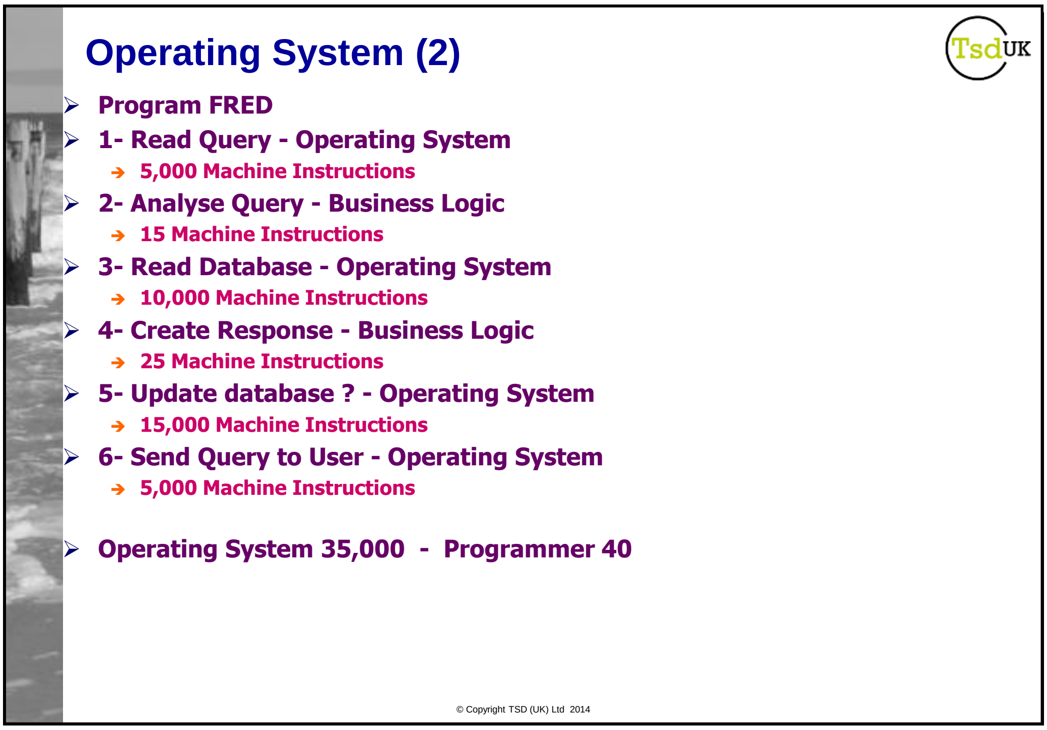
Program FRED

1- Read Query
2- Analyse Query
3- Read Database
4- Create Response
5- Update database ?
6- Send Query to User



Operating System (1)

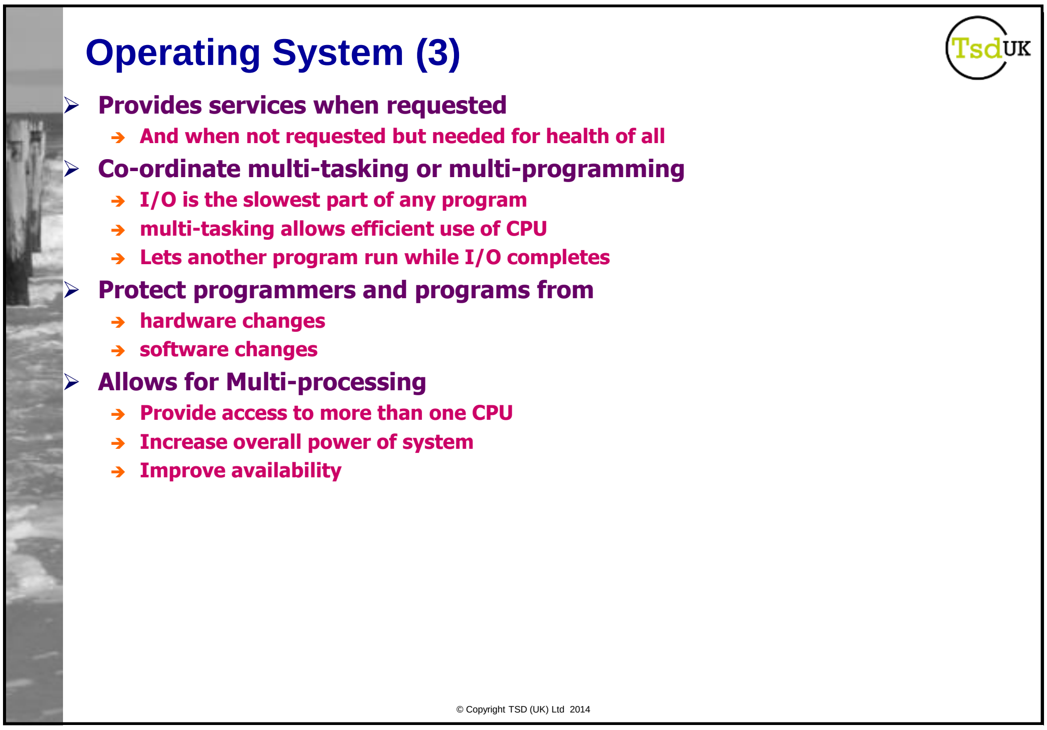
- Programs are simple to code
- Simulate and emulate business logic or process
- The business logic is usage dependent
- Read, Write, Add, etc - frequently the same for many different programs
- So it's much more efficient to do this mundane but difficult stuff once and do it properly
  - → Enter the Operating System !
- Leave programmers to concentrate on Business issues
- Often referred to as the Control Program



Operating System (2)

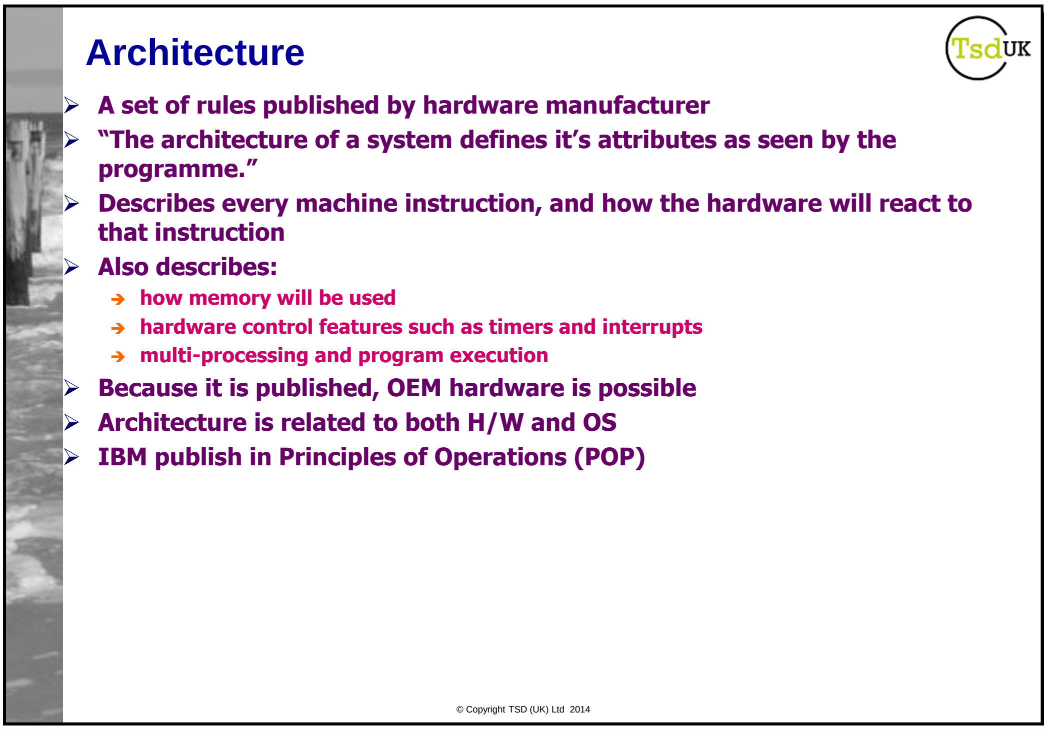
- Program FRED
- 1- Read Query - Operating System
  - → 5,000 Machine Instructions
- 2- Analyse Query - Business Logic
  - → 15 Machine Instructions
- 3- Read Database - Operating System
  - → 10,000 Machine Instructions
- 4- Create Response - Business Logic
  - → 25 Machine Instructions
- 5- Update database ? - Operating System
  - → 15,000 Machine Instructions
- 6- Send Query to User - Operating System
  - → 5,000 Machine Instructions

- Operating System 35,000 - Programmer 40



Operating System (3)

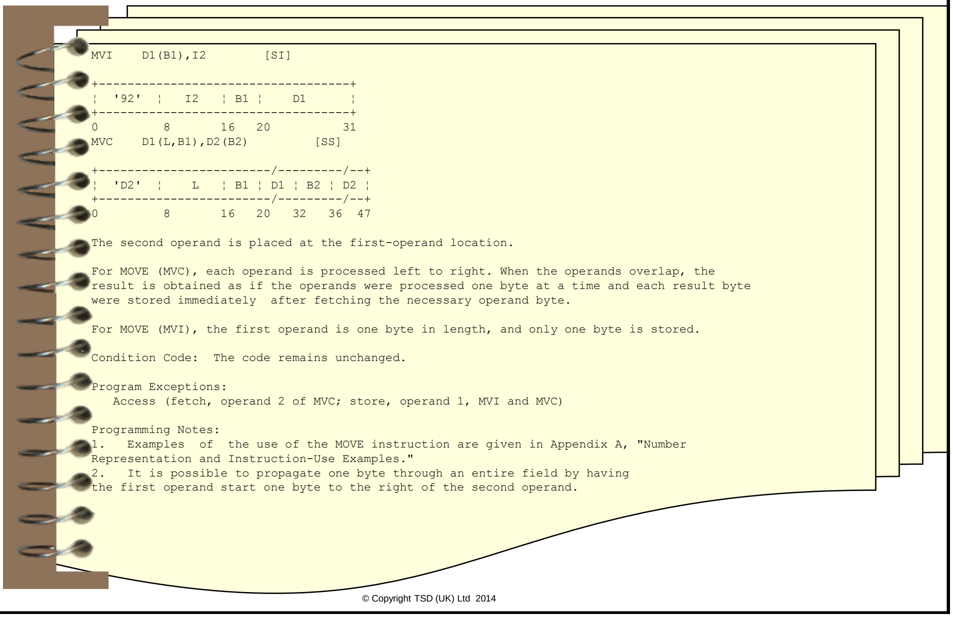
- Provides services when requested
  - And when not requested but needed for health of all
- Co-ordinate multi-tasking or multi-programming
  - I/O is the slowest part of any program
  - multi-tasking allows efficient use of CPU
  - Lets another program run while I/O completes
- Protect programmers and programs from
  - hardware changes
  - software changes
- Allows for Multi-processing
  - Provide access to more than one CPU
  - Increase overall power of system
  - Improve availability



Architecture

- A set of rules published by hardware manufacturer
- “The architecture of a system defines it’s attributes as seen by the programme.”
- Describes every machine instruction, and how the hardware will react to that instruction
- Also describes:
  - how memory will be used
  - hardware control features such as timers and interrupts
  - multi-processing and program execution
- Because it is published, OEM hardware is possible
- Architecture is related to both H/W and OS
- IBM publish in Principles of Operations (POP)

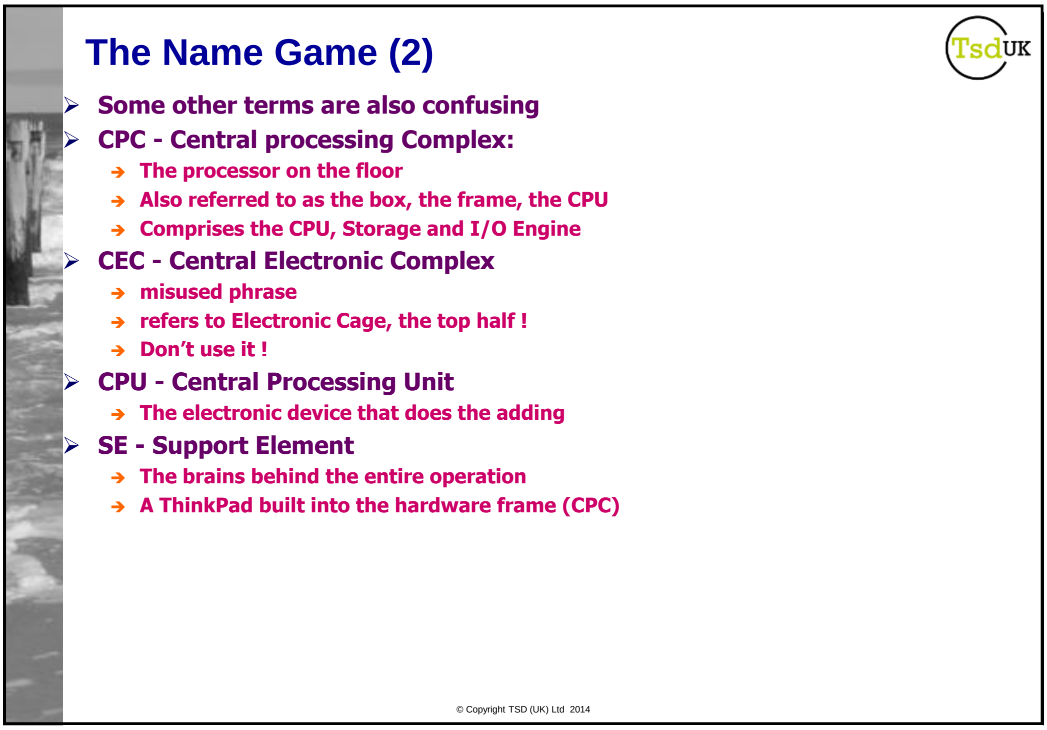
Example of POP - MVC



```
MVI    D1(B1),I2        [SI]

+------------------------------------+
¦  '92'  ¦   I2   ¦ B1 ¦    D1       ¦
+------------------------------------+
0         8       16   20           31
MVC    D1(L,B1),D2(B2)         [SS]

+------------------------/---------/--+
¦  'D2'  ¦    L   ¦ B1 ¦ D1 ¦ B2 ¦ D2 ¦
+------------------------/---------/--+
0         8       16   20   32   36  47

The second operand is placed at the first-operand location.

For MOVE (MVC), each operand is processed left to right. When the operands overlap, the
result is obtained as if the operands were processed one byte at a time and each result byte
were stored immediately  after fetching the necessary operand byte.

For MOVE (MVI), the first operand is one byte in length, and only one byte is stored.

Condition Code:  The code remains unchanged.

Program Exceptions:
   Access (fetch, operand 2 of MVC; store, operand 1, MVI and MVC)

Programming Notes:
1.   Examples  of  the use of the MOVE instruction are given in Appendix A, "Number
Representation and Instruction-Use Examples."
2.   It is possible to propagate one byte through an entire field by having
the first operand start one byte to the right of the second operand.
```

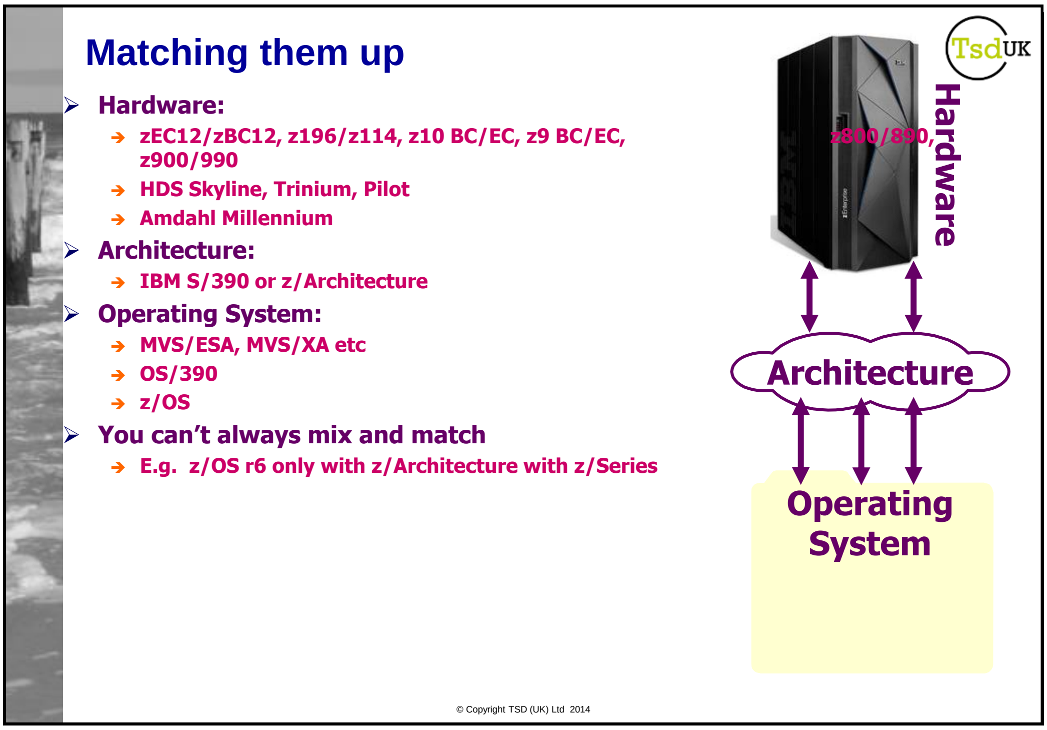
The Name Game (1)

- IBM change names of mainframe components
- Typically when an enhanced product is shipped
- There are many names for IBM's mainframe OS:
  - MVS (MVS/SP, MVS/XA, MVS/ESA)
  - OS/390 r1-10
  - z/OS r1.1 to r1.13 (Sept 2011)
    - z/OS r2.1 announced Sep 2013 – from then on every two years
- This course may use the term MVS when referring to MVS, OS/390 and z/OS
- Many names for the mainframe architecture:
  - S/360, S/370, 370/XA, ESA/390, S/390, z/Architecture
  - All relate to specific hardware devices
  - The OS used with a particular processor must match the hardware
- IBM re-badged all mainframe processors to "zSeries" in 2000
  - Then had to change the name in 2005 to System z
    - Various reasons, mostly related to conflicts with names of other unrelated products
    - i.e. TOTO zSeries Toilet



The Name Game (2)

- Some other terms are also confusing
- CPC - Central processing Complex:
  - The processor on the floor
  - Also referred to as the box, the frame, the CPU
  - Comprises the CPU, Storage and I/O Engine
- CEC - Central Electronic Complex
  - misused phrase
  - refers to Electronic Cage, the top half !
  - Don't use it !
- CPU - Central Processing Unit
  - The electronic device that does the adding
- SE - Support Element
  - The brains behind the entire operation
  - A ThinkPad built into the hardware frame (CPC)



Matching them up

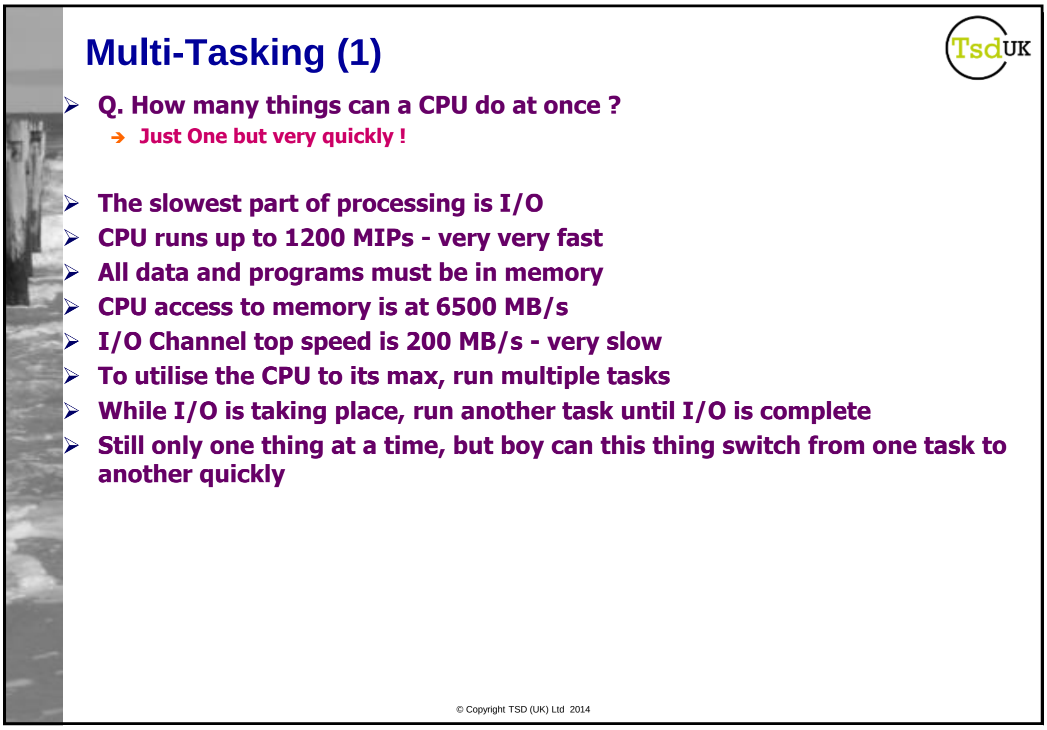
- Hardware:
  - zEC12/zBC12, z196/z114, z10 BC/EC, z9 BC/EC, z900/990
  - HDS Skyline, Trinium, Pilot
  - Amdahl Millennium
- Architecture:
  - IBM S/390 or z/Architecture
- Operating System:
  - MVS/ESA, MVS/XA etc
  - OS/390
  - z/OS
- You can't always mix and match
  - E.g. z/OS r6 only with z/Architecture with z/Series

z800/890,

Hardware

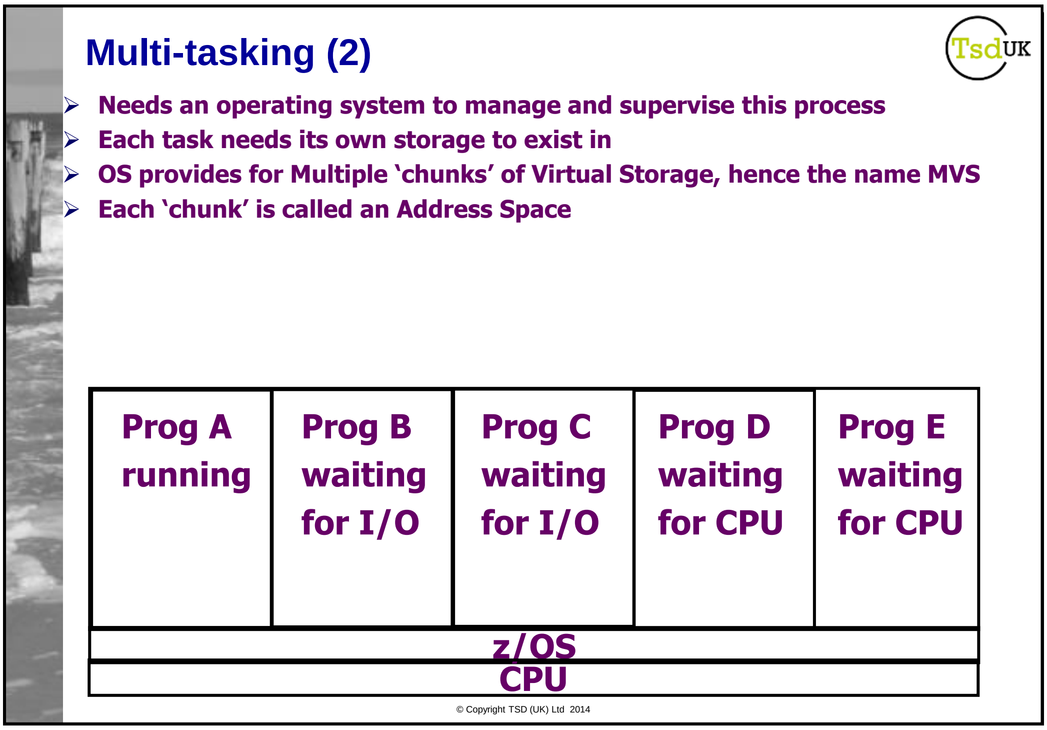
Architecture

Operating System



Multi-Tasking (1)

- Q. How many things can a CPU do at once ?
  - → Just One but very quickly !
- The slowest part of processing is I/O
- CPU runs up to 1200 MIPs - very very fast
- All data and programs must be in memory
- CPU access to memory is at 6500 MB/s
- I/O Channel top speed is 200 MB/s - very slow
- To utilise the CPU to its max, run multiple tasks
- While I/O is taking place, run another task until I/O is complete
- Still only one thing at a time, but boy can this thing switch from one task to another quickly



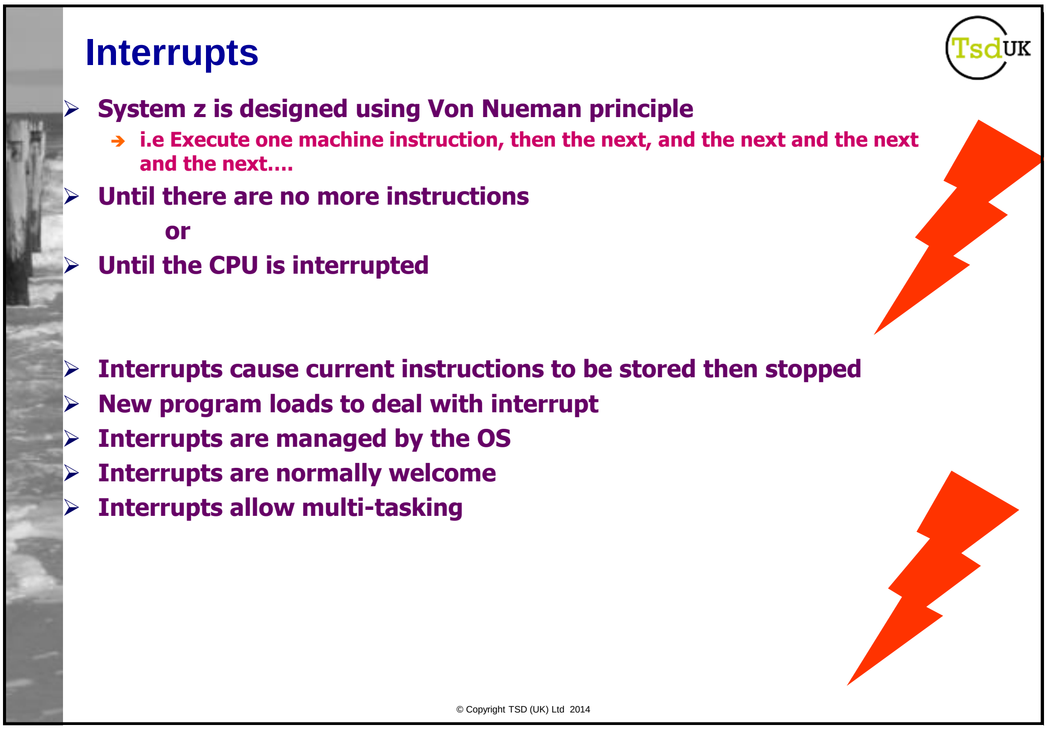
Multi-tasking (2)

- Needs an operating system to manage and supervise this process
- Each task needs its own storage to exist in
- OS provides for Multiple 'chunks' of Virtual Storage, hence the name MVS
- Each 'chunk' is called an Address Space

Prog A running	Prog B waiting for I/O	Prog C waiting for I/O	Prog D waiting for CPU	Prog E waiting for CPU

z/OS

CPU

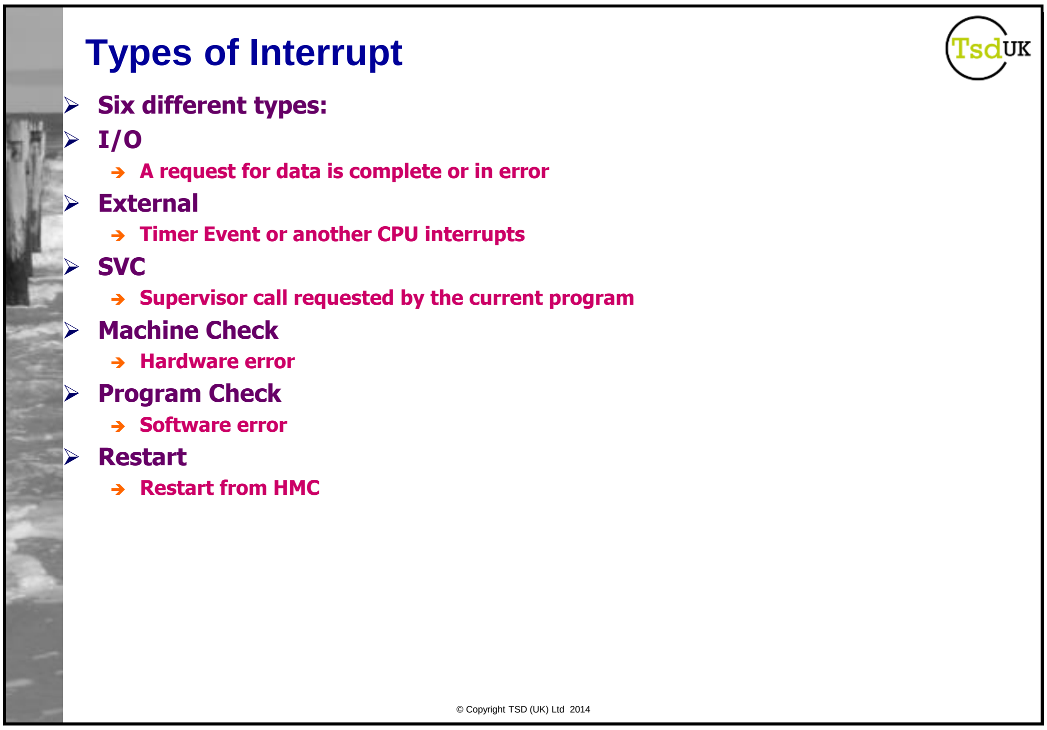


Interrupts

- System z is designed using Von Nueman principle
  - → i.e Execute one machine instruction, then the next, and the next and the next and the next….
- Until there are no more instructions

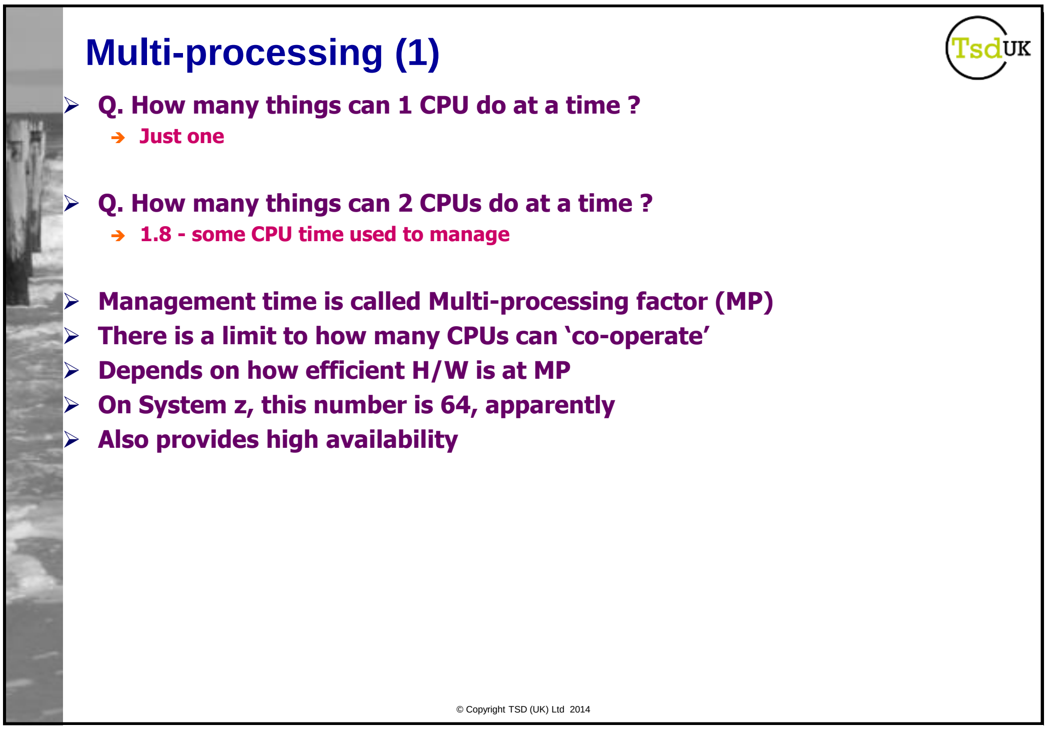
  or
- Until the CPU is interrupted

- Interrupts cause current instructions to be stored then stopped
- New program loads to deal with interrupt
- Interrupts are managed by the OS
- Interrupts are normally welcome
- Interrupts allow multi-tasking



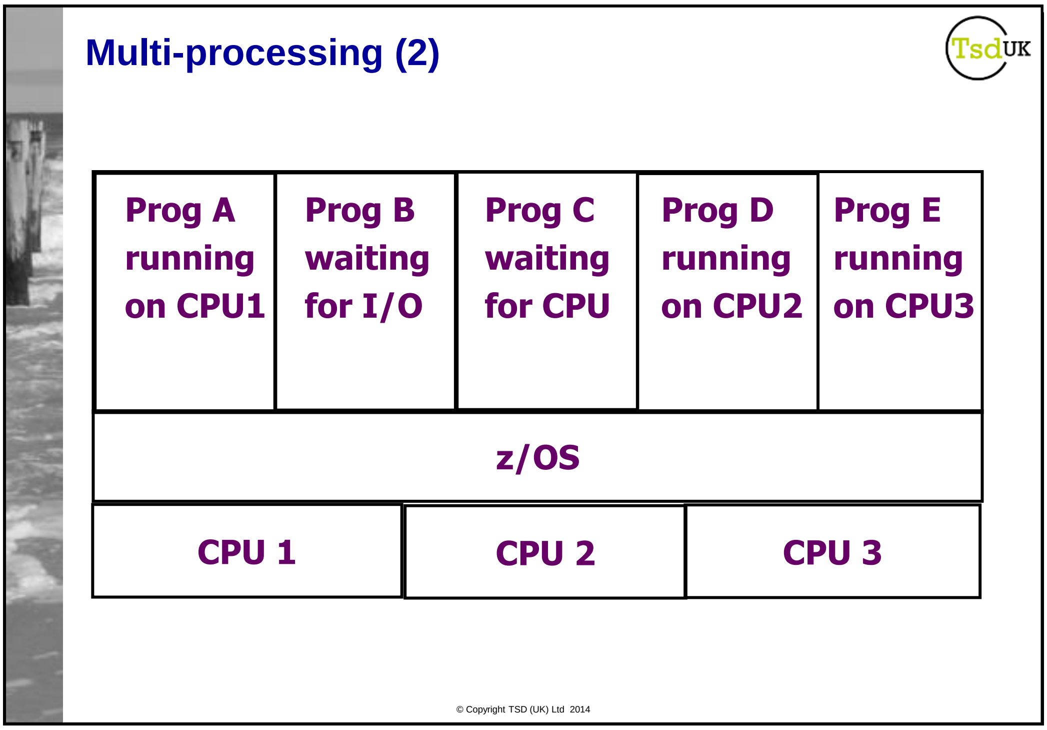
Types of Interrupt

- Six different types:
- I/O
  - → A request for data is complete or in error
- External
  - → Timer Event or another CPU interrupts
- SVC
  - → Supervisor call requested by the current program
- Machine Check
  - → Hardware error
- Program Check
  - → Software error
- Restart
  - → Restart from HMC



Multi-processing (1)

- Q. How many things can 1 CPU do at a time ?
  - → Just one
- Q. How many things can 2 CPUs do at a time ?
  - → 1.8 - some CPU time used to manage
- Management time is called Multi-processing factor (MP)
- There is a limit to how many CPUs can 'co-operate'
- Depends on how efficient H/W is at MP
- On System z, this number is 64, apparently
- Also provides high availability



Multi-processing (2)

Prog A running on CPU1

Prog B waiting for I/O

Prog C waiting for CPU

Prog D running on CPU2

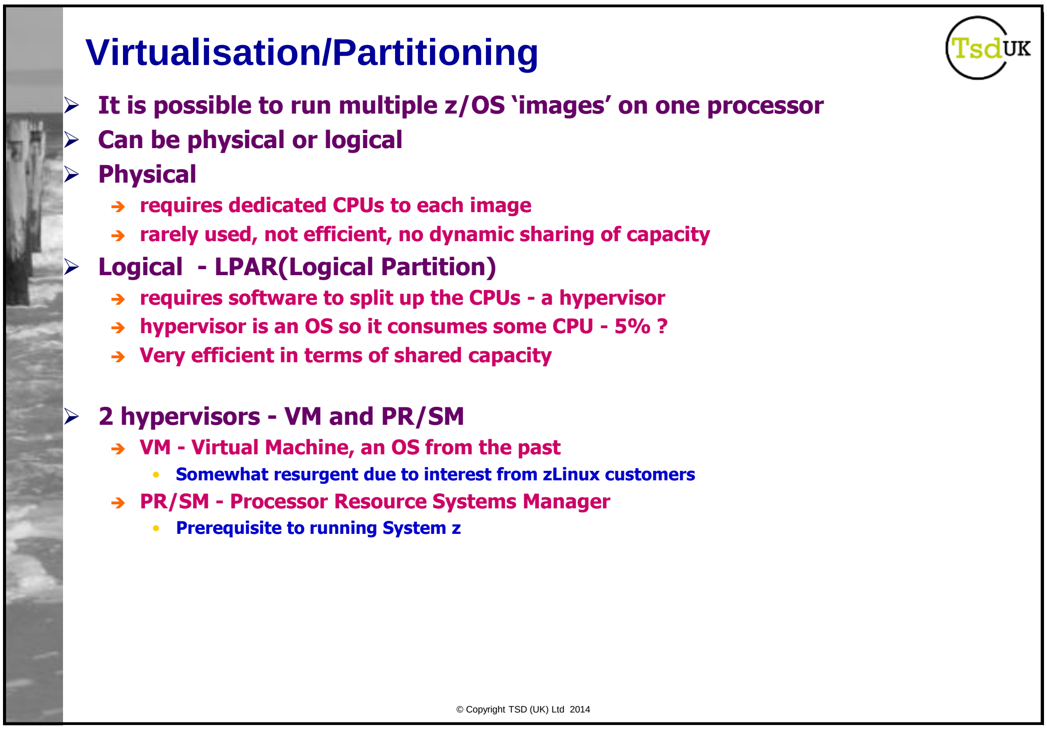
Prog E running on CPU3

z/OS

CPU 1

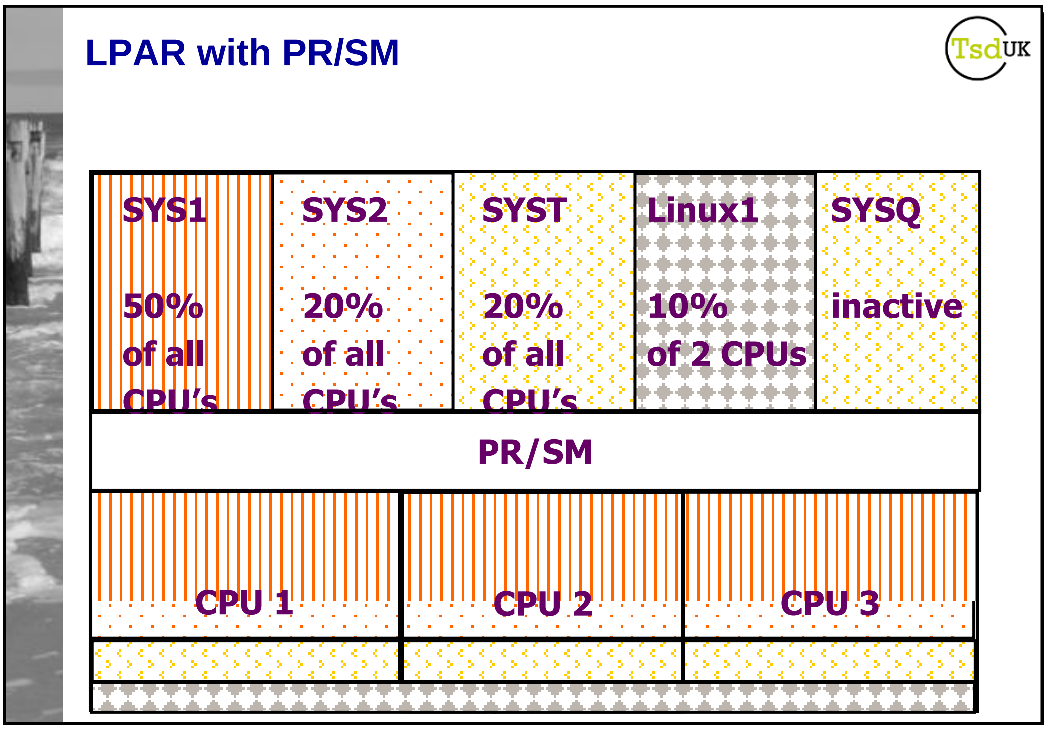
CPU 2

CPU 3



Virtualisation/Partitioning

- It is possible to run multiple z/OS 'images' on one processor
- Can be physical or logical
- Physical
  - requires dedicated CPUs to each image
  - rarely used, not efficient, no dynamic sharing of capacity
- Logical - LPAR(Logical Partition)
  - requires software to split up the CPUs - a hypervisor
  - hypervisor is an OS so it consumes some CPU - 5% ?
  - Very efficient in terms of shared capacity
- 2 hypervisors - VM and PR/SM
  - VM - Virtual Machine, an OS from the past
    - Somewhat resurgent due to interest from zLinux customers
  - PR/SM - Processor Resource Systems Manager
    - Prerequisite to running System z



LPAR with PR/SM

SYS1 50% of all CPU's

SYS2 20% of all CPU's

SYST 20% of all CPU's

Linux1 10% of 2 CPUs

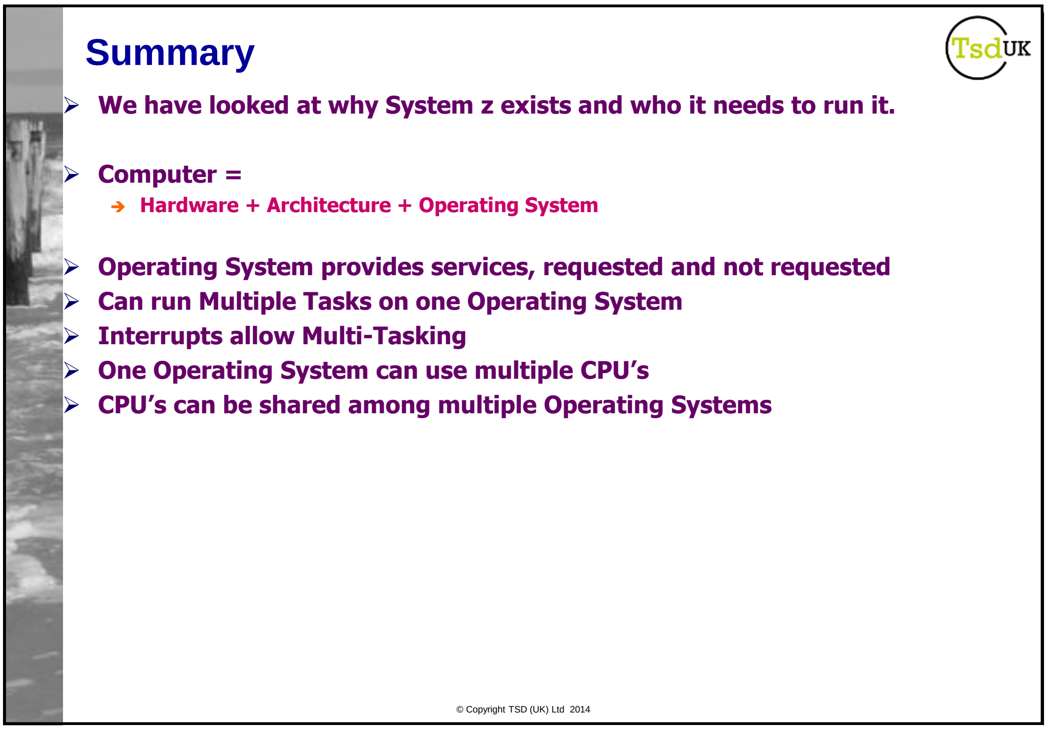
SYSQ inactive

PR/SM

CPU 1

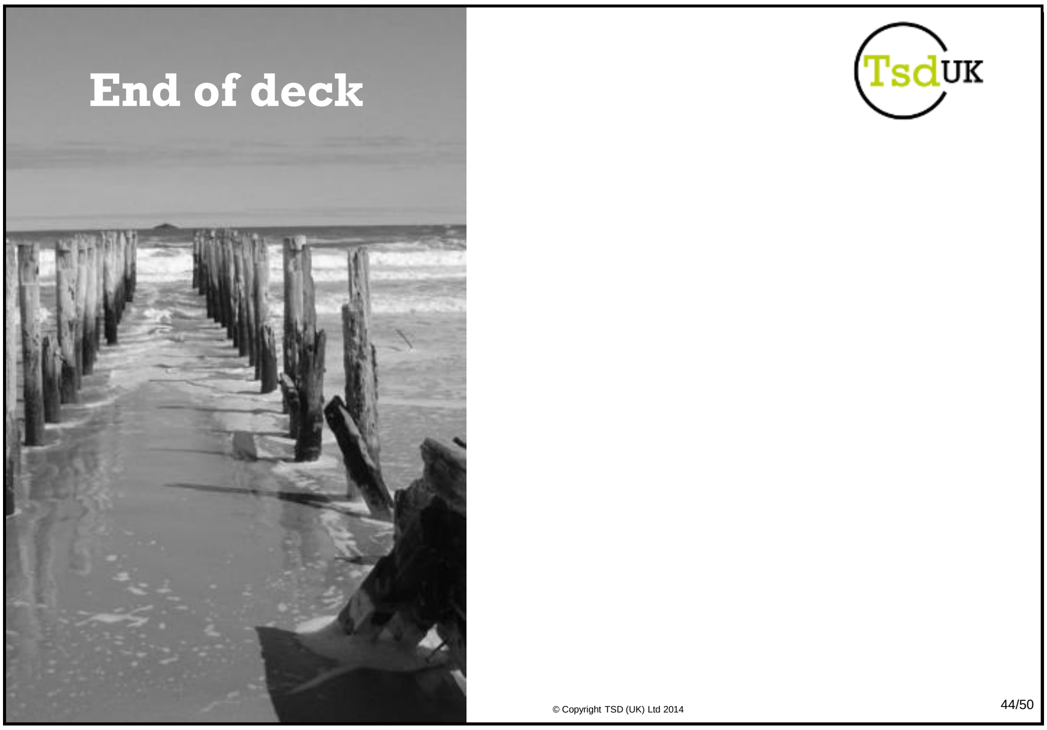
CPU 2

CPU 3



Summary

- We have looked at why System z exists and who it needs to run it.
- Computer =
  - Hardware + Architecture + Operating System
- Operating System provides services, requested and not requested
- Can run Multiple Tasks on one Operating System
- Interrupts allow Multi-Tasking
- One Operating System can use multiple CPU’s
- CPU’s can be shared among multiple Operating Systems



End of deck

44/50