Capability Concept Mechanism and Structure in System 250

By
Christian Diercks

Overview

- Intro
- Plessey System 250 Architecture
- Basic Capability Mechanism
- Basic capability table
- Load Capability Instruction
- Structure of a Package
- Structure of a Program
- Structure of a Resource
- Execution of a program
- Call Return and Store Capability Instruction
- Process Dump Stack
- Structure of a Process
- Structure of a System
- Conclusion
- Reference

Introduction

- Capability concept and implementation
- Plessey System 250
  - first capability system sold commercially
  - First functioning computer to use capability addressing
  - Designed to meet critical real time performance and reliability needs (Tel.)
  - Applying capabilities to a multiprocessor environment

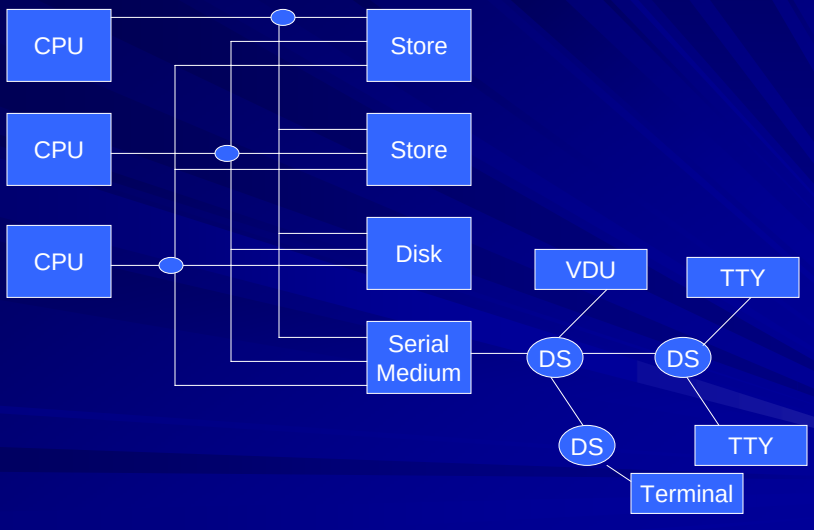
System 250

- Multiprocessor System
  - Symmetric
  - Up to 8 processors with 8 storage modules
  - Segmented memory space by operating system (Virtual Memory)
- Segment contains capabilities or data

System 250

- No event interrupts in system
- Interrupts can occur internally
  - Fault
  - Interval timer
  - Trap

System 250



System 250

- To access data in a memory segment, a program must load one of the capability registers with a capability for that segment
- A Plessey capability permits its possessor to access an object in the system, where an object is a logical or physical resource

Basic Capability Mechanism

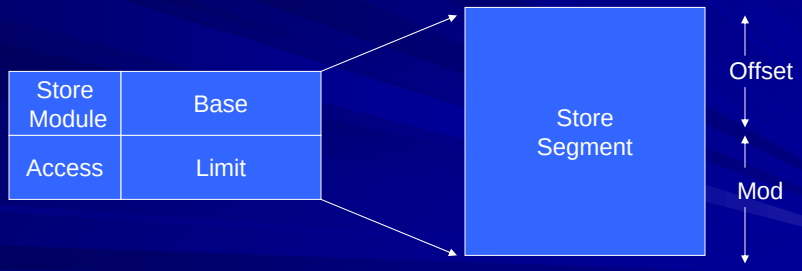
- Definition
  - A capability (also known as a key) is a concept in secure computing. It refers to a value that references an object along with an associated set of access rights
  - A user program must use a capability to access an object

Basic Capability Mechanism

- Provide addressing base for access to segments in fast store
- To Protect a segment against illicit operations
- To limit the scope of the program and thus protect the data structure outside this scope from illicit access

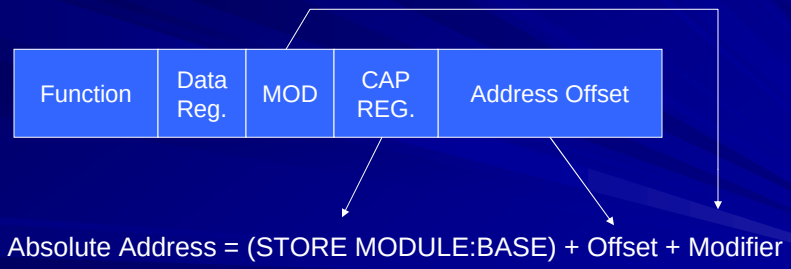
Basic Capability Mechanism

- Addressing Fast Store
- Accessing words with reference to Capability Register



Basic Capability Mechanism

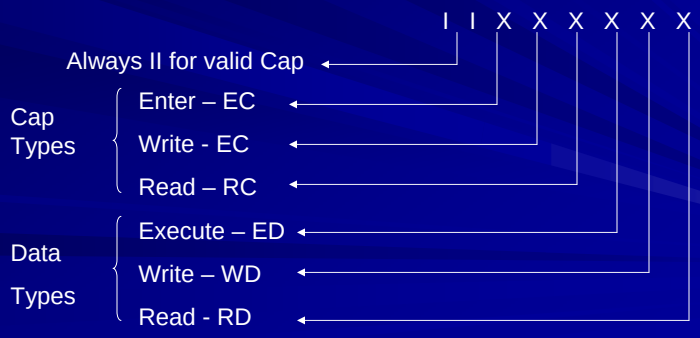
- CPU instructions access words within segment by reference to a capability register which defines it.



Absolute Address = (STORE MODULE:BASE) + Offset + Modifier

Basic Capability Mechanism

- Capability Register define the permitted bounds of a segment and the operations permitted upon it



System Capability Table (SCT)

- Base/limit values defining segments are collected into single segment SCT
- Each processor has an internal register that contains the address of the SCT
- Physical addressing information is centralized and relocation of segments is simplified
- One SCT entry for each object in the system

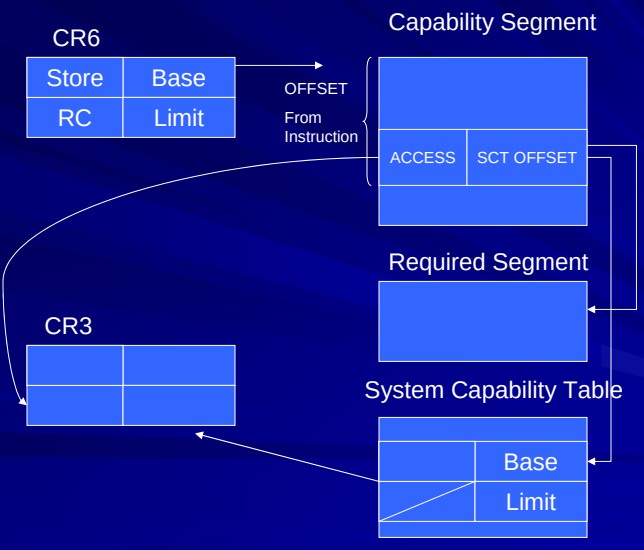
Load Capability Instruction

- A program executes a LOAD Capability instruction to transfer a capability from a capability segment to a capability register.
- Loading capability to get a new segment

Load Capability Instruction

- Steps
  - Hardware examines the SCT index the specified capability in memory
  - Index selects the SCT entry for the segment
  - Capability register is constructed from the right field in the capability and the base and limit from SCT entry

Load Capability Instruction



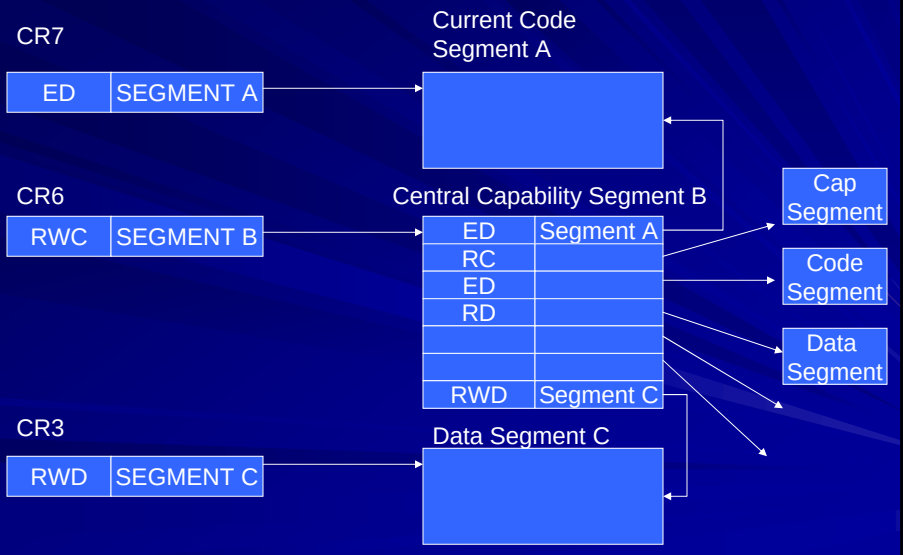
Structure of a Package

- Package consists of a central capability segment that defines a number of satellite segments, which may include further capability segments
- Convention
  - CR(7) defines code segment currently being executed
  - CR(6) defines the central capability segment of the package concerned

Structure of a Package

- A protected subsystem is built by creating a central capability block in which the subsystem will execute
- Central capability block contains capability for code, data, and capability segments available to the executing process

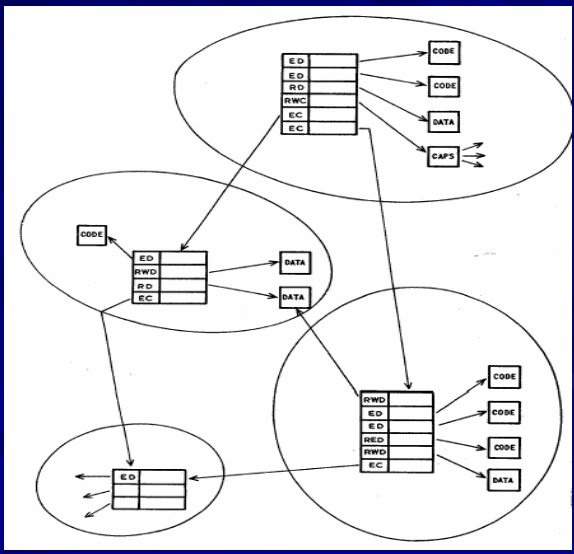
Structure of a Package



Structure of a Program

- A program consists of a number of program packages interconnected by Enter capabilities
- One package being accessible from a number of other others

Structure of a Program



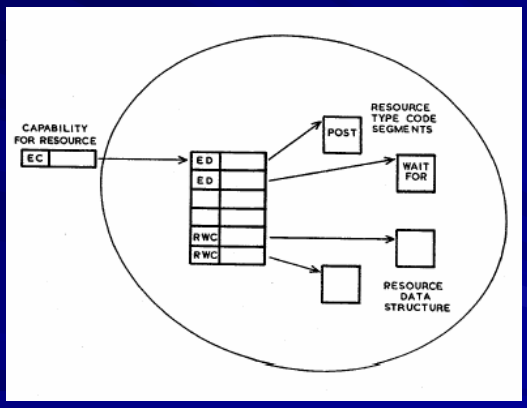
Structure of a Resource

- Plessey 250 operating system is constructed as a set of protected subsystems that manage various types of resources
- Enter type capability ensures no need for privileged mode

Structure of a Resource

- OS provides facilities for allocating and manipulating a number of resource types
  - Store segment
  - Synchronized flag
  - Process
  - Data stream
  - User
  - Job
  - Symbol directory
  - Text file

Structure of a Resource



Execution of a program

- Execution of program may construct a dynamic data structure by repeated calls to resource allocation packages
- Convention
  - CR(5) is used to define the first capability segment of the process data structure

Call Return and Store Capability Instructions

- Subroutine calls are performed by call instructions
  - Enter type capability for package central capability block
  - Offset to the execute type capability of the code segment to be entered
- Effect of call instruction
  - Load execute type capability into CR(7)
  - Load enter type capability into CR(6)

Call Return and Store Capability Instructions

- Two further actions are required
  - To operate on the data structure a read capability access type needs to be supplied
  - Before CR(7), CR(6) can be overwritten, their old values need to be preserved for return instructions

Process Dump Stack

- Each process is associated with a segment called “Process dump stack”
- Process dump stack contains two parts
  - Stack area for preservation of CR(6), CR(7), and IAR values during a call instruction
  - A dump area in which remaining register values can be preserved on interrupt or context change

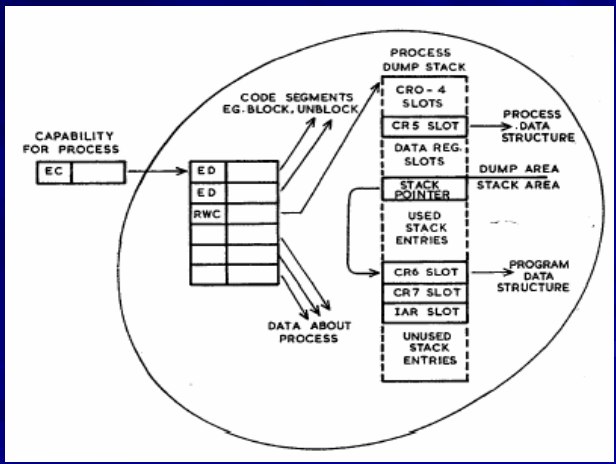
Structure of a Process

- Central capability segment of the process defines a number of segments which contain general information about the process
- A process which creates another is supplied with an enter type capability

Structure of a Process

- Central Capability segment of the process defines a number of segments which contain general information about the process
- One of the segments defined is the process dump stack

Structure of a Process



Structure of the System

- Many processes simultaneously
- Virtual processor for each process
- CPU scheduler
- Change Process instruction
- Providing complete control from one process to another by change process instructions
- Dump stack preserves old process information

Conclusion

- Plessey 250 uses capabilities to simplify MP
- Capabilities aid software error detection
  - Each process possesses capability for only those segments needed for its function
- The Plessey System 250 combines hardware and software support to provide a uniform view of system resources

Reference

- http://www.informatic.uni-ulm.de/rs/projecte/monads/capabilitiesE.html
- http://www.cs.washington.edu/homes/levy/capabook/Chapter4.pdf
- http://www.cs.ucf.edu/%7Eeurip/cop6614/englandplessey250.pdf