
Capability-Based Addressing

Paper by

R.S. Fabry (Univ. Of California)

Presented By

Rajan John Thomas

Overview

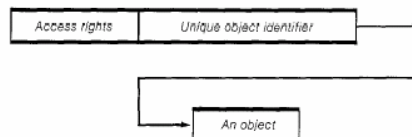
- Problem Overview
 - Introduction
 - Shared Address Solutions
 - Implementation
 - Integrity
 - Address Translation
 - Conclusion
 - Reference
-

Problem Overview

- The Problem
 - Addressing schemes using segment table
 - Inadequacy with shared address
- Solution using Capabilities
 - Use of capability as an address solves this problem
 - Use of tags for capability
 - Main memory - used as a hash table
 - Recently accessed address are looked up using associative registers rather than main memory

Introduction

- What is Capability ?
 - Conceptually a token, ticket or key that gives the processor permission to access an entity or object in the computer system.
 - Implemented as a data structure that contains unique object identifier and access rights.



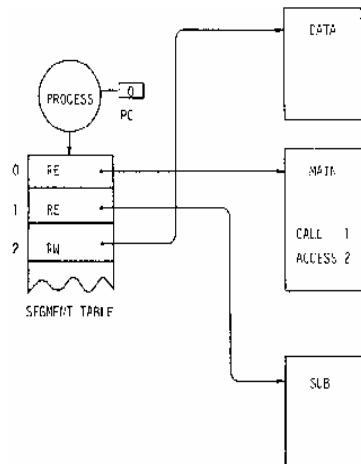
Benefits of Capability

- Single mechanism to address both primary and secondary memory.
- Expanded to all objects in the system rather than only to memory.
- Special kind of address that can be created only by the system
- Can be used as a protection mechanism

Context Independent Addresses

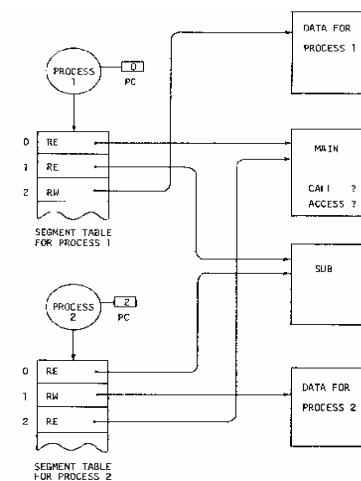
- Capability provides absolute address for an object
- Static Relocation
 - Initially there was no address relocation.
 - Only load time relocation was allowed.
 - Resulted in under utilized computers.
- Dynamic Relocation
 - Address relocation was allowed
 - The virtual objects can be relocated as far as the relation between the object and the capability is maintained.

Segment Reference



- An example using Segment Table
- Segment table contains access information
- Main program
 - calls a subroutine - CALL1
 - accesses a data segment - ACCESS2

Shared Segment Address

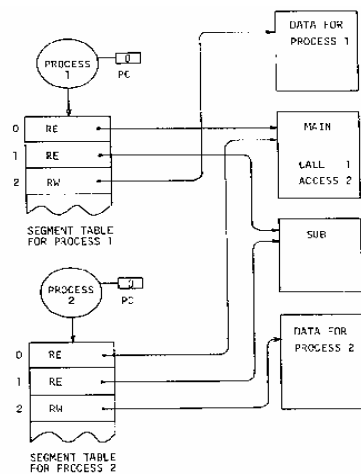


- Uses same code for execution with different data segments per process
- What should the CALL number be ?

Shared Segment - Solutions

- Uniform Address Solution
- Indirect Evaluation Solution
- Multiple Segment Table Solution
- Capability Addressing Solution
 - Our area of focus
- Other Solutions

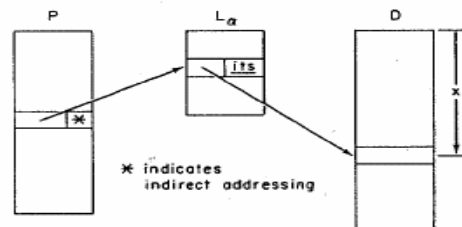
Uniform Address Solution



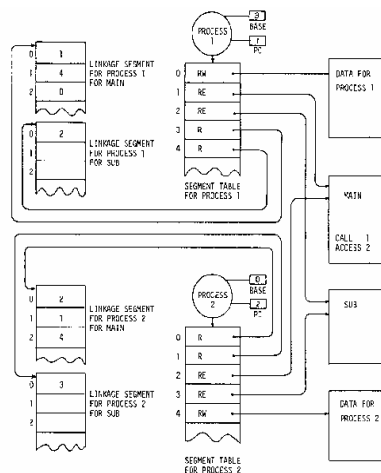
- Functions should be centrally defined
- Used in Burroughs Systems
- Compile all programs at once - compiler allocates segment address at compile time

Linkage Segment

- Indirect reference
- Location through which an external reference may be accessed.
- Symbolic names of each externally known symbol within this segment



Indirect Evaluation Solution

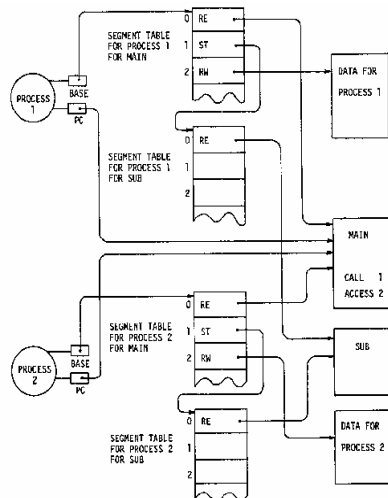


- Shared segment address treated as an index of a position within a linkage segment
- One Linkage segment per subprogram per process.
- Segment table are process wide.
- Main-Call1-Main's Linkage Segment – Value 4 (Segment table) – Sub's Linkage Segment – Value 2 – Looked up under segment table

Indirect Evaluation Solution (Contd)

- Disadvantages
 - Extra space to hold the indirection information.
 - Extra overhead to setup the indirection
 - Extra memory references to fetch indirection.
 - No provision for address used by many process and many programs.
- Used in Multics systems

Multiple Segment Table Solution

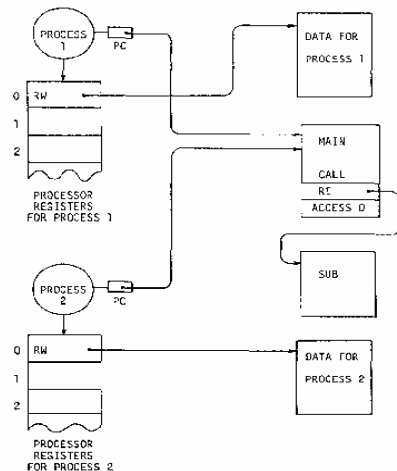


- Linkage segments - replaced by capabilities.
- Linkage segments are renamed segment tables.
- Base register and program counter modified to contain capabilities.

Multiple Segment Table Solution ...

- Segment table are private to particular program as well as to a particular process.
- Disadvantages
 - Parameter passing is difficult
 - Disallows recursive subroutine or requires a new version of segment table for every recursive level.

Capability Addressing Solution



- Capabilities may be stored in segments and in registers of processors.
- Subroutine call follows the capability for the segment containing the subroutine to be called.
- The allocation of processor register is controlled by the compiler.

Capability Addressing Solution ...

- The register can be redefined by storing the contents and later restoring them.
- No requirement for a centralized mechanism to define the use of registers.
- Reference to sub-routine refers to the same segment for each process and is embedded directly in the program

Other Solutions

- Unique Integer Solution
 - Address each segment with a unique integer
 - Assigned at the time the segment is created.
 - Not re-used when the segment is removed
 - Access rights must be determined separately.

Relative vs Absolute Address

- Multics - Absolute addressing is used
 - Every segment is referred by the full path name of the system
 - Protection provided by per segment basis (R/W/E/A)

 - Modern Multiprogramming
 - Rely on Shared Objects.
-

Hardware Implementation

- Chicago Magic Number Computer
 - University of Chicago
 - System 250
 - Plessey Company
 - Use capability for protection at the Operating System Level.
 - Tagged Machines
 - Burroughs B6700, Rice Computers
-

Integrity Of Capabilities

- No normal program can alter the bit pattern that is used to represent a capability.
- Approaches
 - Tagged approach
 - Burroughs B6700
 - Rice Computers
 - Partition approach
 - Chicago Magic Number Machine
 - Plessey System 250 (future presentation)

Tagged approach

- Tagged approach
 - A bit is added to each word in the segment to denote whether it is a capability or not.
 - Data - Information that is not a capability
 - testing/setting of the bit done by the processor on each access.
 - The copied word is given the same bit.
 - Addressing always checks whether the segment address is tagged as a capability or not.

Burroughs Descriptor Format

Tag	P	Drum number	Segment size	Drum address	Memory address
-----	---	-------------	--------------	--------------	----------------

Data and Program Descriptor

Tag	P	Unit number	Operation size	Operation type	Format/control	Memory address
-----	---	-------------	----------------	----------------	----------------	----------------

I/O Descriptor

Partition Approach

- Partition Approach
 - Separation between capability and data
 - At create time each segment is designated as either capability or data
 - one set of process registers for capability and data.
 - Data can be copied to and from a data registers only.
- Partition and Tagged approach are equivalent

Address Translation

- Capability is simply an address for a virtual object
 - specified whenever the object is accessed.
- Capability is a bit pattern
 - Indication to the address translation logic to locate the virtual object
- Chicago Magic Number
 - in-form capability - Represents Main Memory Reference
 - out-form capability - Represents Secondary storage address.

Address Translation (Contd...)

- Future Implementation of Capability
 - unique code associated with each segment
 - assigned during creation and retains until deleted.
 - Hash table kept in the main memory by the Operating System
 - Contains the unique key and the presence bit (present in main memory)

Hash Table Implementation

- Lookup of the Hashtable
 - Data present in Hashtable with present bit set (present in Main Memory)
 - Data present in Hashtable with present bit not set (type A exception)
 - Data not present in the Hashtable (type B exception)
- An associative memory is used for recent accessed addresses
 - Not required to go to the main memory.

Conclusion

- Capability based addressing
 - Efficient type of absolute address for an object
 - Simplifies programming conventions for shared addresses.
 - Offers
 - protection
 - simplicity of programming convention
 - efficient implementation

References

- Programming Semantics for Multiprogrammed Computations
 - Jack B Dennis & Earl C. Van Horn - March 1966
 - Communications of the ACM 9(3), March 1974, pp. 143-155.
- Capability-Based Computer Systems
 - Book by Henry M. Levy
 - <http://www.cs.washington.edu/homes/levy/capabook/>

Thank You