On the Duality of Operating System Structures

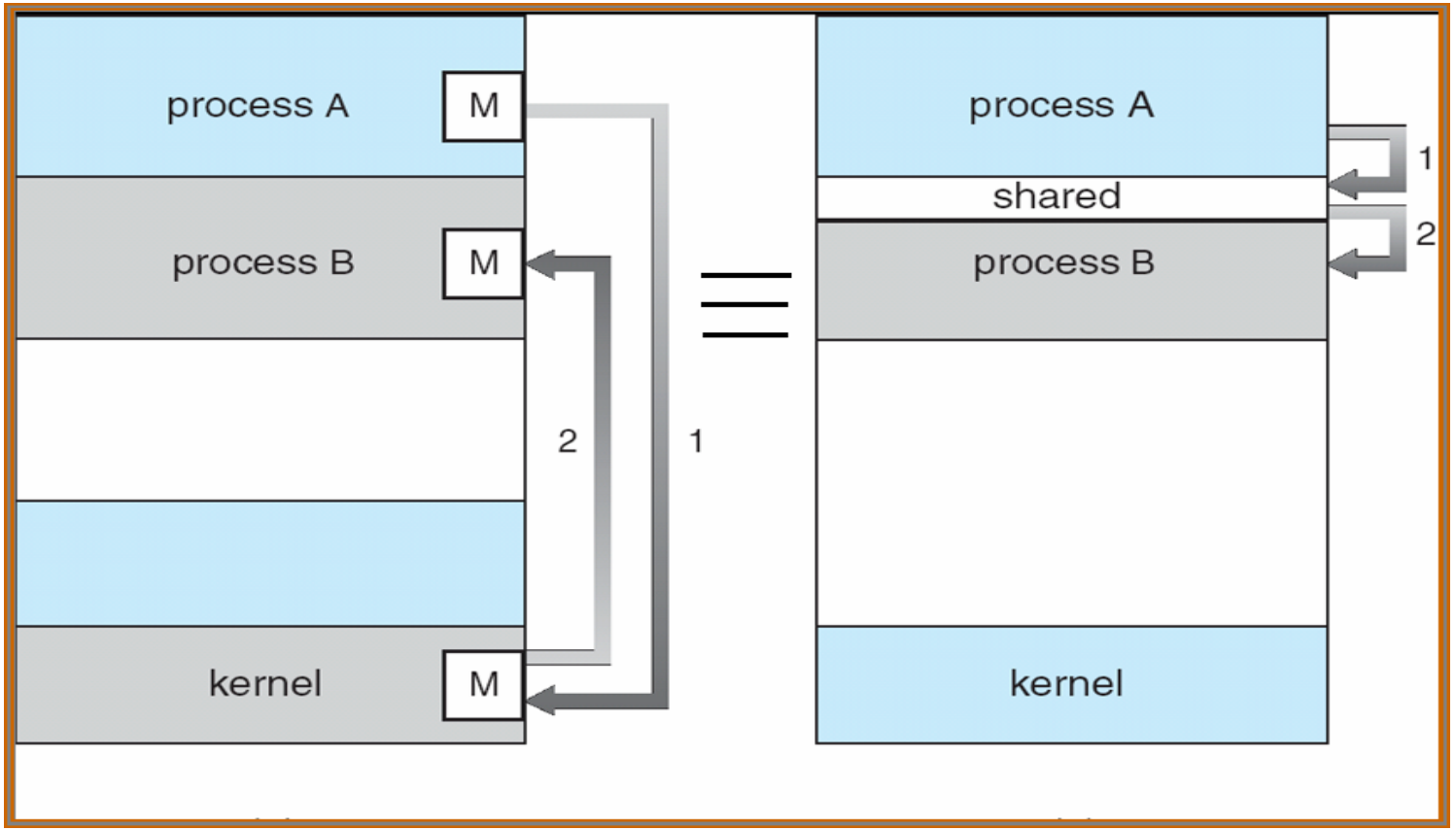
- Lauer, Needham

Presented by – Kapil Ahuja

Key Point

Message-oriented system

Procedure-oriented system



* Image from OSC

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Basics (Terminology)

- In Paper (1970's)
- Today (2000's)

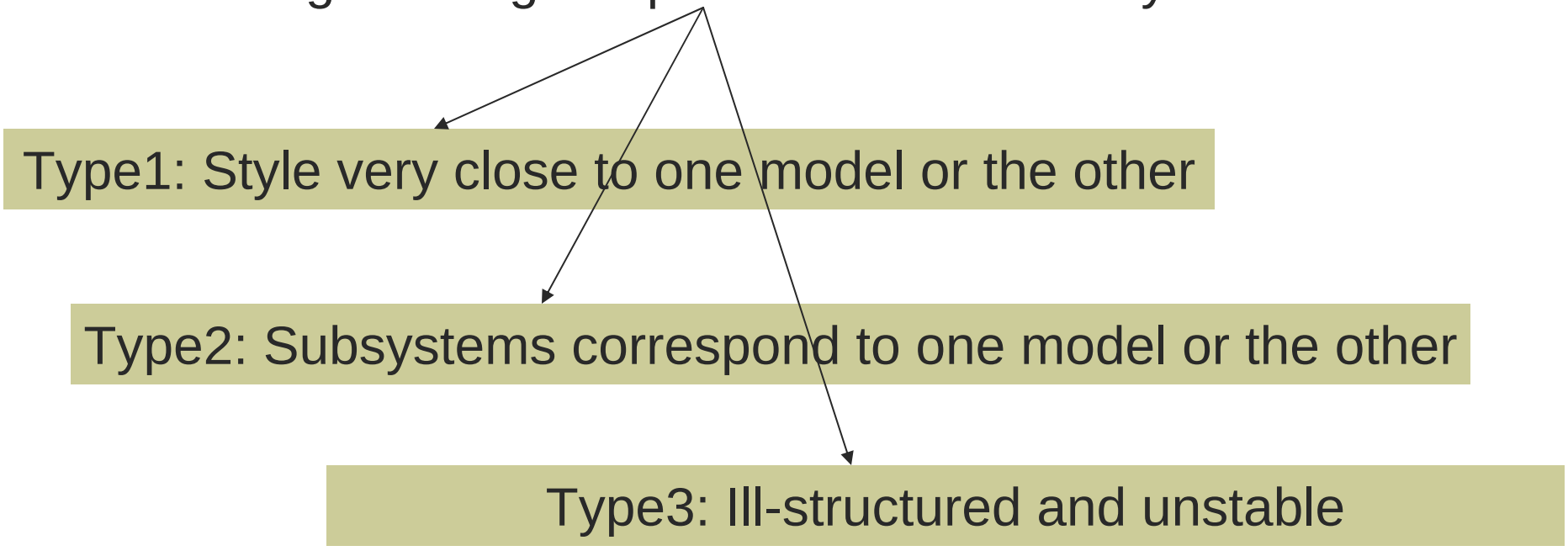
Message Oriented	Event Based
Procedure Oriented	Thread Based

- Mapping not exact as events today use:
  - Cooperative multitasking (basically non-preemptive multitasking)
  - Shared memory

These not present in message oriented system of paper

Basics (Introduction)

- Most OS and internet servers can be classified using message or procedure oriented system



Basics (Objectives)

- Eliminate uninformed controversy about which is “better” to build. In general:
  - Message oriented – simpler concurrency model
  - Procedure oriented – simpler & natural programming style
- Eliminate several degrees of freedom in the design process

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Message Oriented System

- Characterized by:
  - Small number of (relatively static) big processes
  - Explicit set of message channels
  - Limited amount of direct sharing of data in memory
- Examples:
  - Real-time systems
  - General OS: IBM's OS/360, GEC 4080

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Procedure Oriented System

- Characterized by:
  - Large number of very small processes
  - Rapid creation and deletion of processes
  - Communication by means of direct sharing of data in memory
- Examples:
  - HYDRA
  - Plessey System 250

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Obv. 1 & 2: Duality Mapping & Similar Programs

- Message-oriented system
- Procedure-oriented

message ports	procedure identifiers simple
SendReply	RETURN (from procedure) monitor
…	…
…	…
…	…
…	…
…	…
SendMessage; AwaitReply (immediate)	procedure call
SendMessage;... AwaitReply (delayed)	FORK; . . .JOIN

….Obv. 1 & 2 (Contd.) – Server's

- Message-oriented

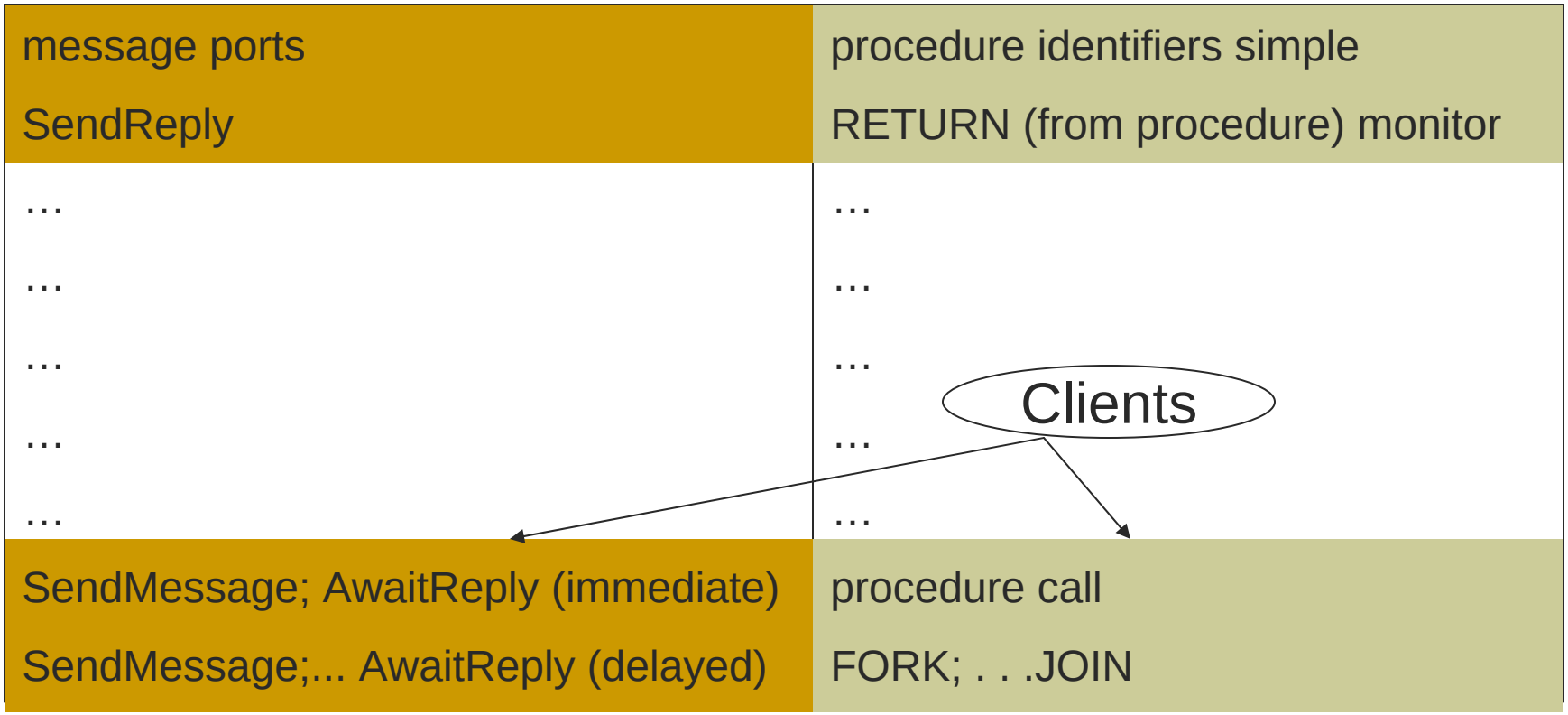
```
begin m: messageBody
        i:messageId, p:portId, s:set of portId
        resourceExhausted: boolean flag

        do forever
                [m, i, p] =  WaitForMessage[s]
                case p of
                        port 1 =>...

                        port 2 =>...
                                if resourceExhausted then
                                        s = s - port2;
                                SendReply[i, reply];
                                …

                        port L =>.
                                s = s + port 2
                                ...
                 endcase
        endloop
end
```

- Procedure-oriented

```
ResourceManager: MONITOR =
        C: CONDITION
        resourceExhausted: BOOLEAN

        proc 1: ENTRY PROCEDURE[…] =…

        proc 2: ENTRY PROCEDURE[...] RETURNS[…] =
                BEGIN
                        IF resourceExhausted THEN
                                WAIT C
                        RETURN [results]
                        ...
                END

       proc L: ENTRY PROCEDURE[...] =
                BEGIN
                        resourceExhausted = FALSE
                        SIGNAL C
                        …
                END
END
```

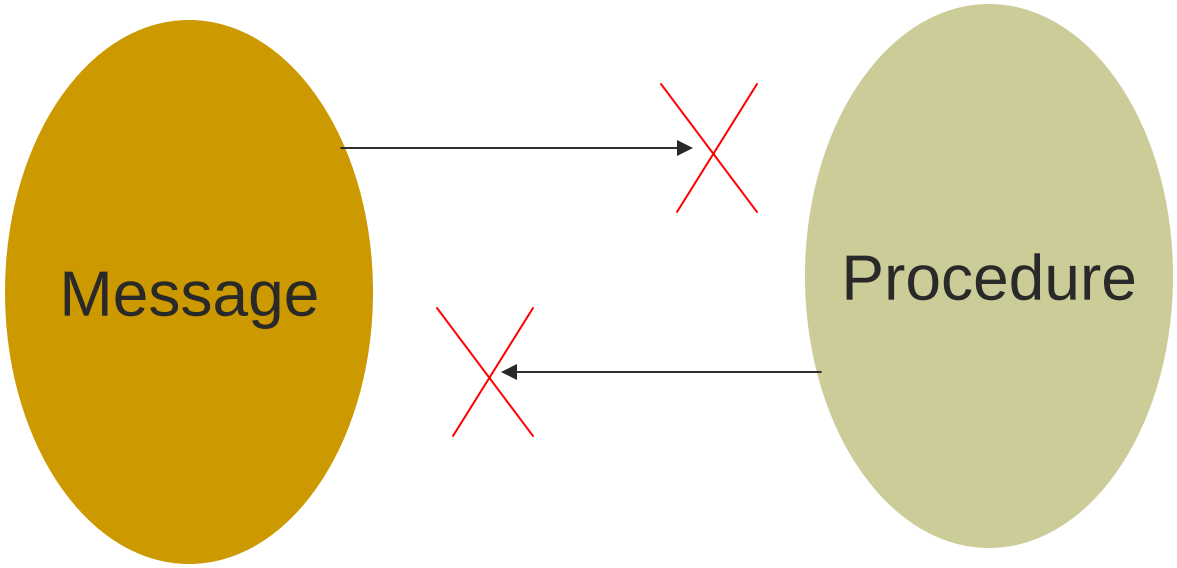
Obv. 1 & 2: Duality Mapping & Similar Programs

- Message-oriented system
- Procedure-oriented



….Obv. 1 & 2 (Contd.)

- Open Question



Is having a no reasonable counterpart a good thing?

Tracker

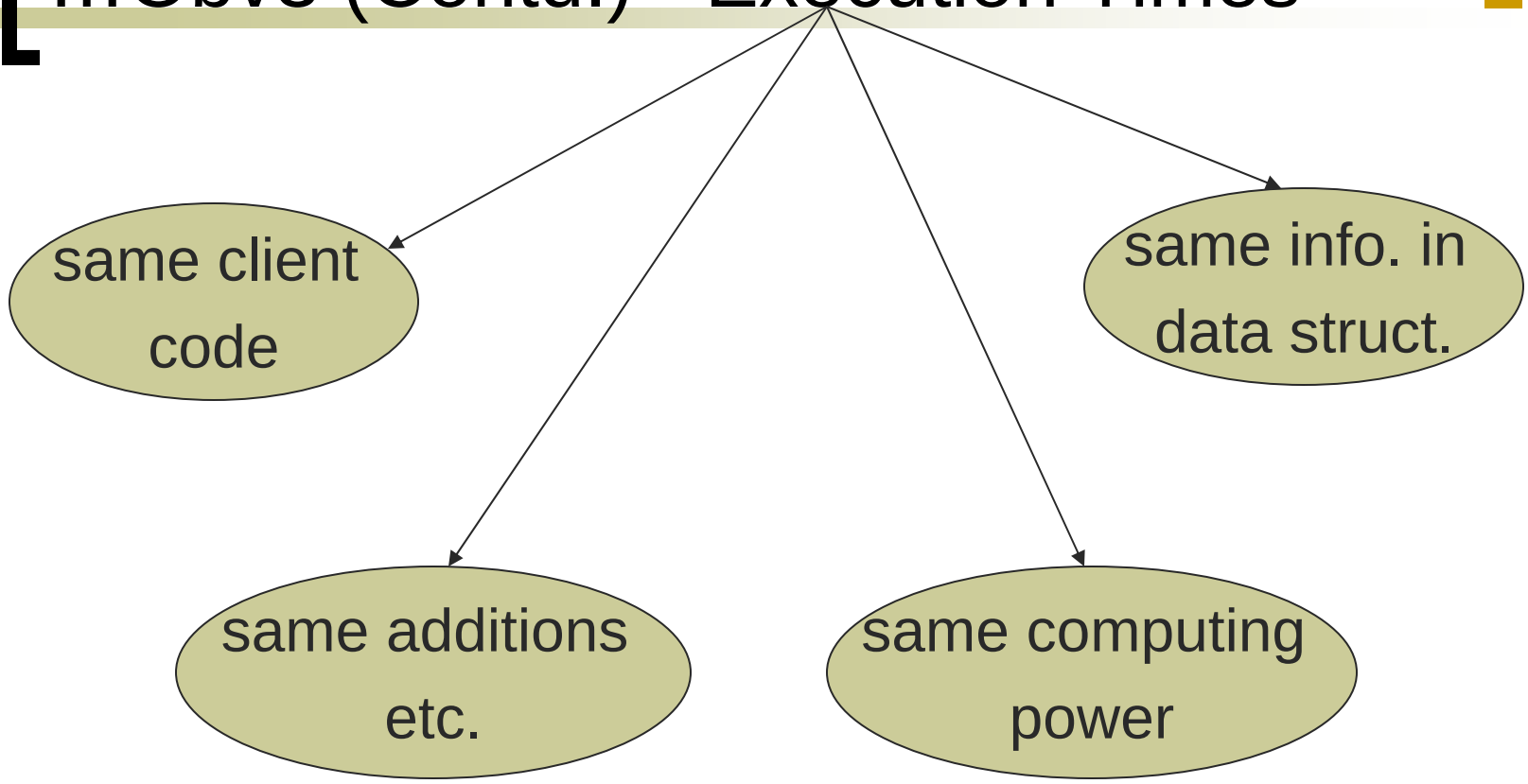
- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Obv3: Performance Preservation

- 3 components of the dynamic behavior:

1. Execution times of programs themselves
2. Computational overhead of primitive system operations
3. Queuing and waiting times reflecting congestion and sharing of resources

…Obv3 (Contd.) - Execution Times



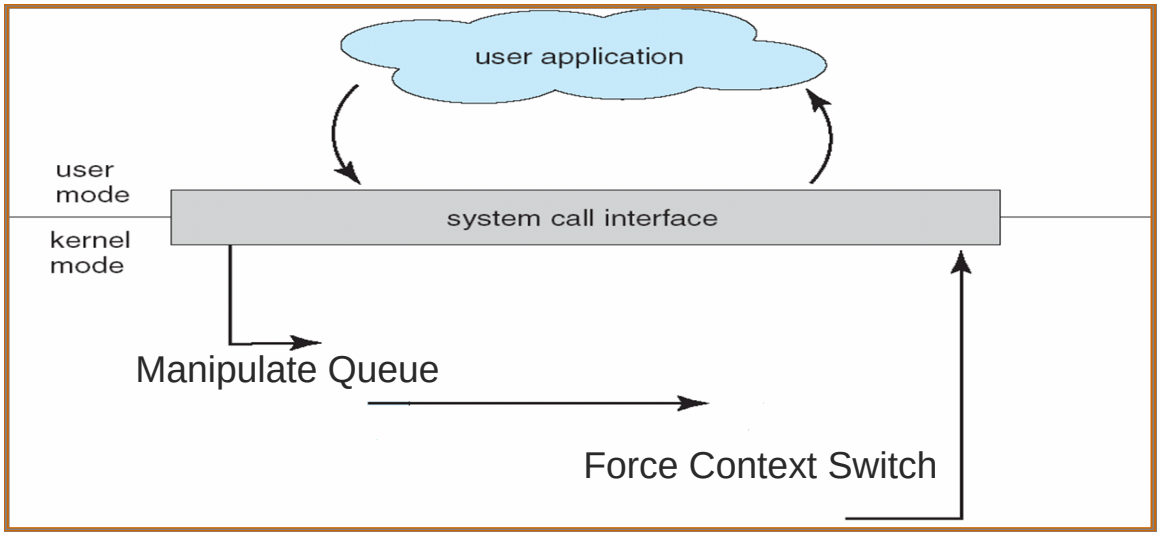
Obv3: Performance Preservation

- 3 components of the dynamic behavior:

1. Execution times of programs themselves
2. Computational overhead of primitive system operations
3. Queuing and waiting times reflecting congestion and sharing of resources

…Obv3 (Contd.) – Comput. Overhead

- This implies the background things can be made equally efficient.
- Example: Message oriented – Send Message OR
  Procedure oriented – Call a procedure



* Image from OSC

Obv3: Performance Preservation

- 3 components of the dynamic behavior:

1. Execution times of programs themselves
2. Computational overhead of primitive system operations
3. Queuing and waiting times reflecting congestion and sharing of resources

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Finally: Which One to Use?

- Depends on the substrate upon which the system is built
- Basically the following criteria's:
  - Organization of real & virtual memory
  - Ease of scheduling and dispatching
  - Arrangement of peripheral devices & interrupts
  - Architecture of instruction set & programmable registers
- Thus advantages to have a system in which changing from one form to other is easy

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Should you Change?

- Not easy to change to reflect the suggested duality
- Why?
  - Underlying addressing structures etc. tightly bound to the design
  - Transformation to a dual version not justified by the second order gains
- Example where easy to change:
  - Cambridge CAP Computer

Tracker

- Basics
- Message Oriented System
- Procedure Oriented System
- Obv.1 & 2: Duality Mapping & Similar Programs
- Observation3: Performance Preservation
- Finally: Which One to Use?
- Should you Change?
- Conclusion

Conclusion: Still Controversial!

- It was a empirical study i.e. no rigorous proofs
- Thus, number of people still disagree to this duality

My Evaluation: Summary

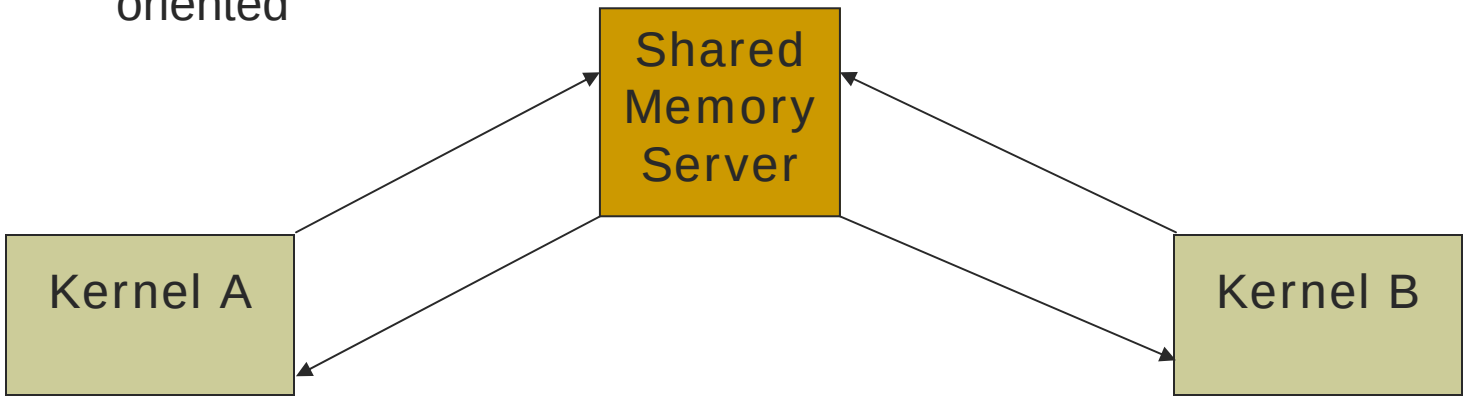
- “It's alright -- you're both doing ok, and you're not that different.”
- In modern times:



…….up to a constant factor of crashes.

My Eval.: Intercomputer Comm.

- Message oriented system preferred
- Why?
  - Easier to implement
- How?
  - No troubles like the shared memory server as in procedure oriented



References

- Papers:
  - On the Duality of Operating System Structures - Lauer, Needham
  - Why Events Are A Bad Idea (for high-concurrency servers) - Rob von Behren, Jeremy Condit and Eric Brewer
  - SEDA: An Architecture for Well-Conditioned, Scalable Internet Services - Matt Welsh, David Culler, and Eric Brewer
- Books:
  - Operating System Concepts – Silberschatz, Galvin, Gagne
  - Modern Operating Systems – Tanenbaum
- Others:
  - Summary by Jonathan Ledlie at Harvard University
  - Presentation by David Allen at Portland State University
  - Presentation by Mehmet Belgin at Virginia Tech

Questions?