

Background

Microkits: Chip level experiments with some "intelligent" control. Drifted into "misuse" for CS2 assembly language practicals.

Fred Machines: Advanced microkits for board-level experiments with more sophisticated control ("boot" processor). Drifted into "misuse" as general computing resource - "APM".

"Proper" operating system was not designed in from start because not originally intended as GCR.

New System

Do we need one? [Filestores, Comms protocols, Users?]
 Do we want one? [of Unix]
 Ought we to have one? [Systems]

Attributes

- Multi-process (obviously)
- Multi-processor (desirable, in keeping with original aims)
- Closely-coupled environment (common memory)

Mouse Kernel

Solid foundation on which to build rest of system - nothing fundamentally new. Kernel viewed as agent which performs a set of actions on a set of objects.

- | | |
|--|---|
| <ul style="list-style-type: none"> - Objects - queueable - processes - messages - queues - run queues - wait queues (semaphores) - mailboxes | <ul style="list-style-type: none"> - Actions - signal ("V") - wait ("P") - send - receive - (internal) sub-actions - lock/unlock - enqueue/dequeue - re/un/in/ex-queue |
|--|---|

Kernel is processor independent - Supervisor is not

Supervisors implement kernel facilities and others (scheduling, interrupts)

Inter-supervisor data interface

Need to define memory layout (record format) of kernel objects

Doubly-linked lists

- ease of arbitrary manipulation, diagnosability
- cf danger of inconsistency, queues "always empty"

```

+-----+
|Tag field |
+-----+
|Forward (Head) |
+-----+
|Backward (Tail)|
+-----+
|Header      |
+-----+
| ...      |
    
```

Semaphore: Counter, Associated Semaphore
 Mailbox: Semaphore
 Run Queue: Tickler (NB no interrupt wires)
 Message: Reply mailbox, Identity of sender, application-specific part
 Process: Target RQ, Async mailbox, General mailbox, processor-specific part

(eg 68010: SSP, XVT, TVT, timers, scheduling, memory management, environment)

Byte Sex

Numbering (addressing) of smaller units within larger units

IBM/Motorola	DEC/Intel
76543210	76543210
+-----+-----+-----+-----+	+-----+-----+-----+-----+
Byte 0 Byte 1 Byte 2 Byte 3	Byte 3 Byte 2 Byte 1 Byte 0
+-----+-----+-----+-----+	+-----+-----+-----+-----+
Halfword 4 Halfword 6	Halfword 6 Halfword 4
+-----+-----+-----+-----+	+-----+-----+-----+-----+
Fullword 8	Fullword 8
+-----+-----+-----+-----+	+-----+-----+-----+-----+
31 0	31 0

Loosely-coupled systems usually go for "byte-fidelity". In closely-coupled systems there is a case for supporting "word-fidelity" because of the large number of pointers. Yet byte-fidelity is ALSO wanted.

	MB0	MB1	MB2	MB3	
PB0	----- 1	----- 9	----- 5		Mode-bit decides whether to enable switches 1234 (for byte fidelity) or 5678 (word fidelity)
PB1	----- A	----- 2	----- 6		
PB2	-----	----- 7	----- 3	----- B	Alternative method might opt to provide halfword fidelity as well (switches 9ABC).
PB3	----- 8	-----	----- C	----- 4	

Naming - Dictionaries

- Standard interrupt handler registration mechanism
- Basic language support, standard & custom libraries
- Loading (dynamic)
- I/O modules
- Memory management (VM essential)