Project groups will be sent out this afternoon

Start working on your project proposal (due 10/13)

HW8 is due a week from now (10/11)

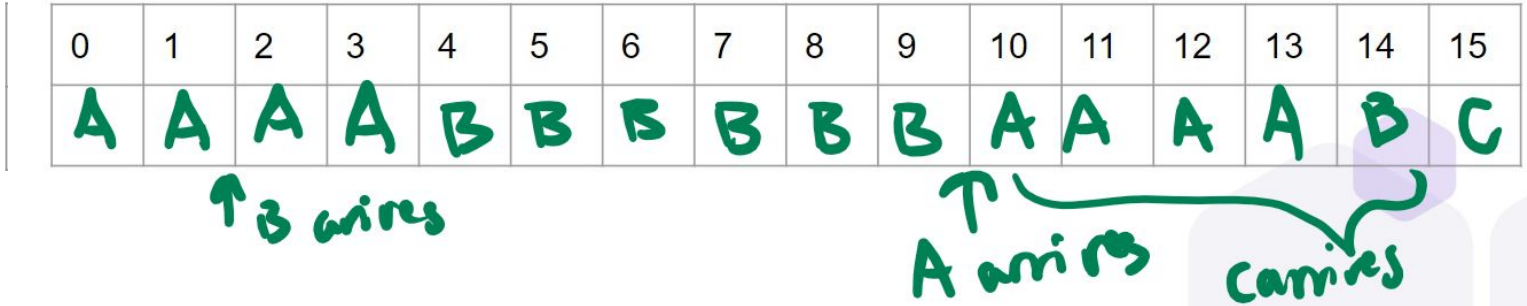
13: Scheduling and RTOS

Latency and priority

High priority interrupt: A (4 ms every 10 ms)

Lower priority interrupts: B (7 ms every 100ms), C (1ms every 15 ms)

Can C fail to execute within 15 ms?



Scheduling

Decide when CPU runs what task so that deadlines are met

Soft: correctness "degrades" if deadlines aren't met

vs

Hard: correctness fails if deadlines aren't met

Example: interrupts

Dynamic: done at run-time

vs

Static: done at compile-time

Preemptive: task can interrupt lower-priority task

vs

Non-preemptive: tasks can't interrupt each other

Example: cyclic execution

Feasibility of scheduling periodic tasks

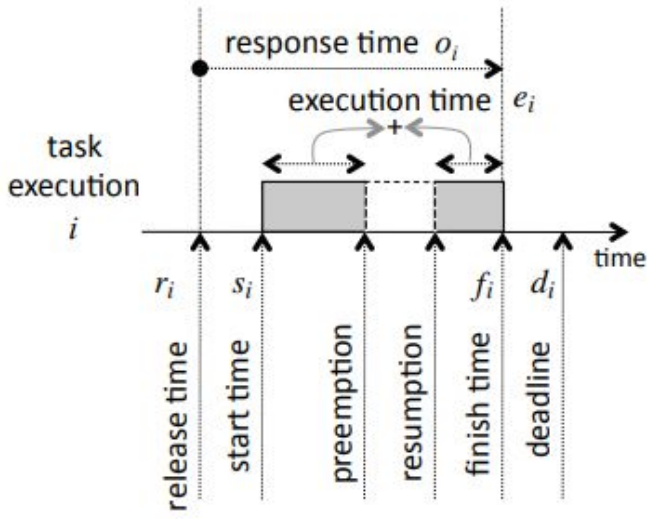
Feasibility: feasible if $f_i \leq d_i$ *for all i*

Utilization: % of time CPU spends executing tasks (vs idle)

Necessary but not **Sufficient** condition for feasibility:

Sum of e_i/p_i (aka e_i/d_i) for all *i* is at most 1

Aka utilization <= 100%



latency: si - ri

Scheduling examples on the board

Dynamic scheduling:

- EDF (earliest deadline first) – schedule next task based on deadline
- LSF/LLF (least slack/laxity first) – schedule next task based on how much execution it has left to do

Wednesday, October 5, 2022 11:13 AM

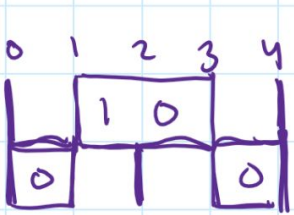
non-preemptive, static

	E	P
T_1	2	4

✓

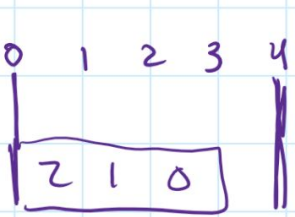
$\sum e/p = 2/4 = 50\%$

	E	P
T_1	2	4
T_2	1	2



$\sum e/p = \frac{2}{4} + \frac{1}{2} = 100\%$

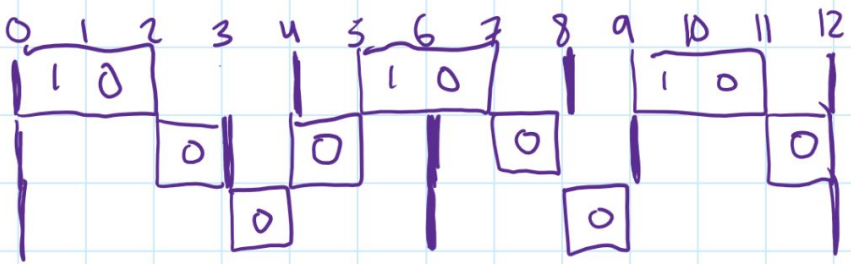
	E	P
T_1	2	4
T_2	3	4



$\sum e/p = \frac{2}{4} + \frac{3}{4} = 125\%$

Still non-preemptive, static

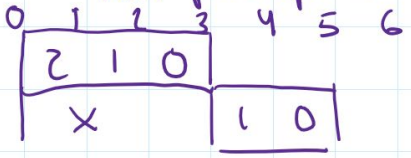
	E	P
T_1	2	4
T_2	1	3
T_3	1	6



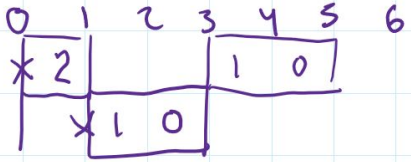
non-periodic EDF (dynamic)

	Release	Exec.	Deadline
T_1	0	3	6
T_2	1	2	5

non-preemptive:



preemptive:



non-preemptive:

	Release	Exec.	Deadline
T_1	0	4	5
T_2	2	6	15
T_3	5	4	13

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

preemptive:

(still according to EDF)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

T_1

T_2

T_3

Rate Monotonic Scheduling (RMS)

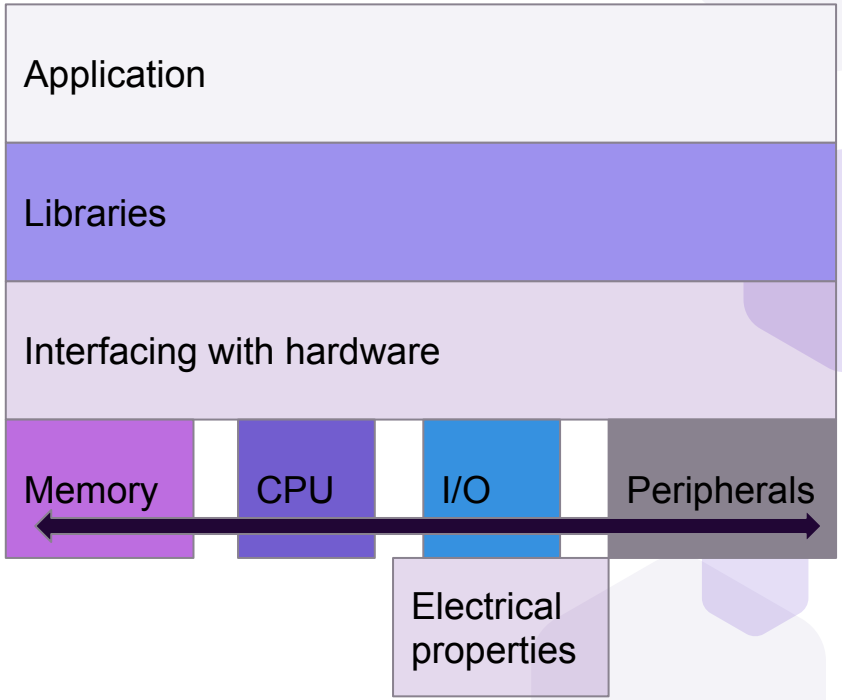
Fixed-priority, determined ahead of time

- Each task has its own priority
- Task with smallest period = highest priority
- Pre-emptive (higher priority tasks interrupt lower-priority tasks)

Guarantee of scheduling when utilization < 69.3%

$$\mu \leq n(2^{1/n} - 1), \qquad (12.2)$$

Stepping back – Embedded systems as systems



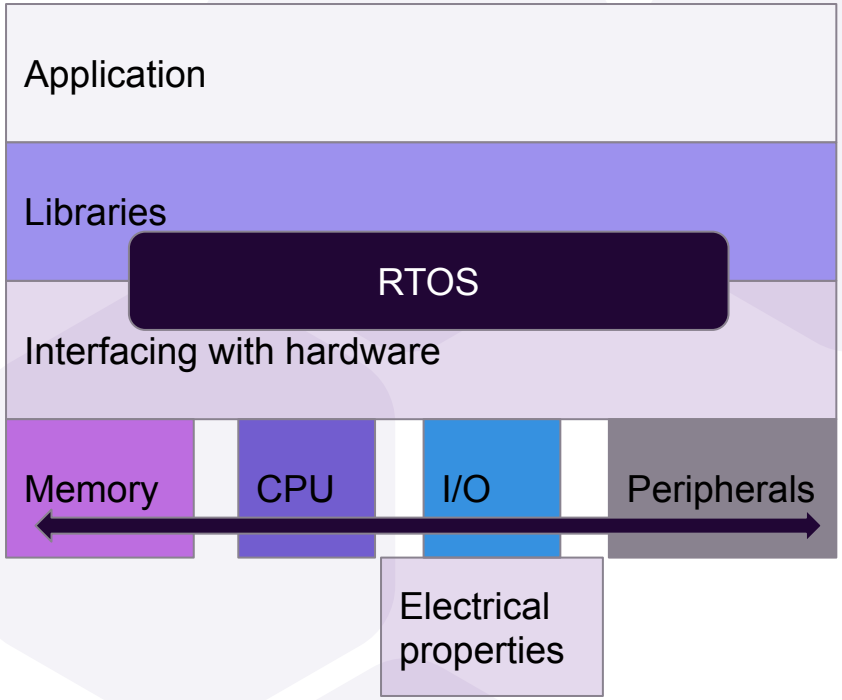
Real-Time Operating Systems

OS - manages system resources and provides services to programs/processes/threads

RTOS - an OS with real-time constraints

- Scheduling policies
- Often support for prioritization
- Libraries for mutexes/semaphores
- Memory management

freeRTOS scheduler description


Application
Libraries
RTOS
Interfacing with hardware
Memory
CPU
I/O
Peripherals
Electrical properties

Pros/cons to using an RTOS?

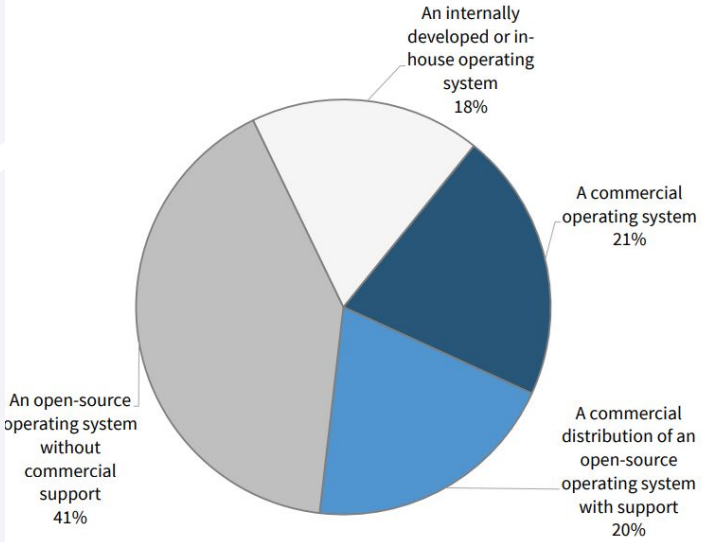
Would you want to write your own RTOS?

Most embedded projects utilize an operating system

Although open source is popular, four in ten use either commercial OS or open-source OS distributed commercially

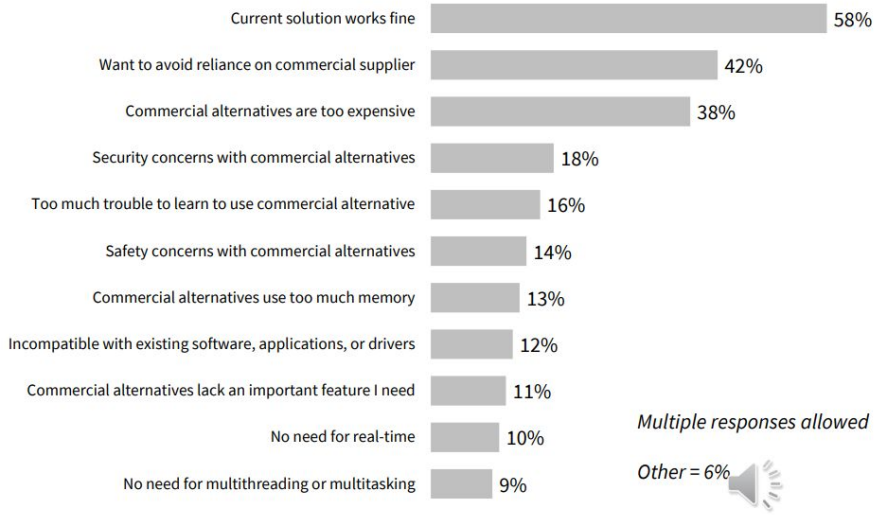
74% use an OS in current embedded project

OS Used in Current Embedded Project



Total Respondents

Reasons for not using commercial OS



Multiple responses allowed

Other = 6%

Base = Those not using commercial OS (284)

embedded survey

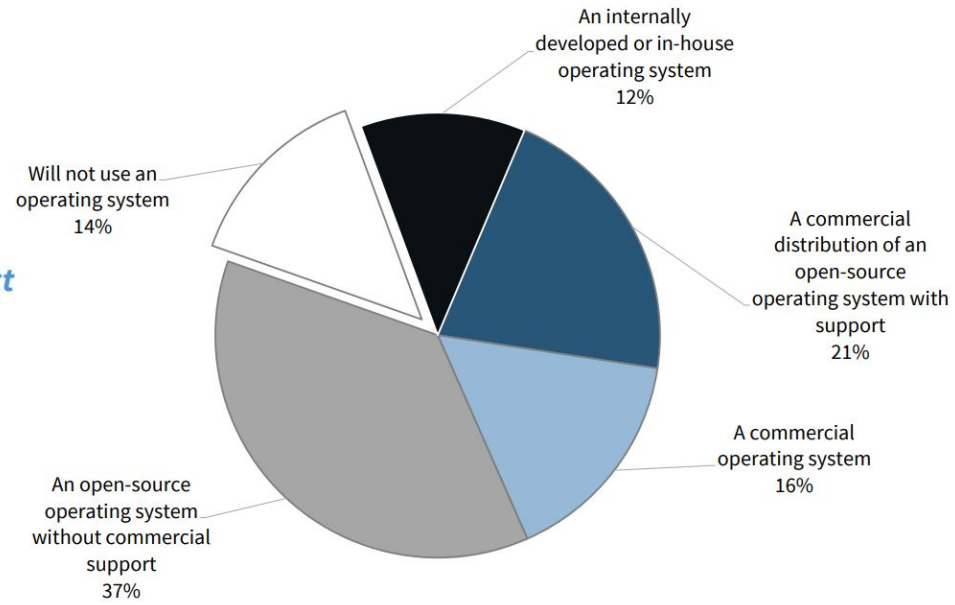
29. Does your current embedded project use an operating system, RTOS, kernel, software executive, or scheduler of any kind?
30. My current embedded project uses:
31. 31. What are your reasons for NOT currently using a commercial operating system?

Image source

OS use will increase, but open-source share will grow

Nearly 30% of those now using commercial OS are considering open-source alternatives

86% will use an OS for next embedded project

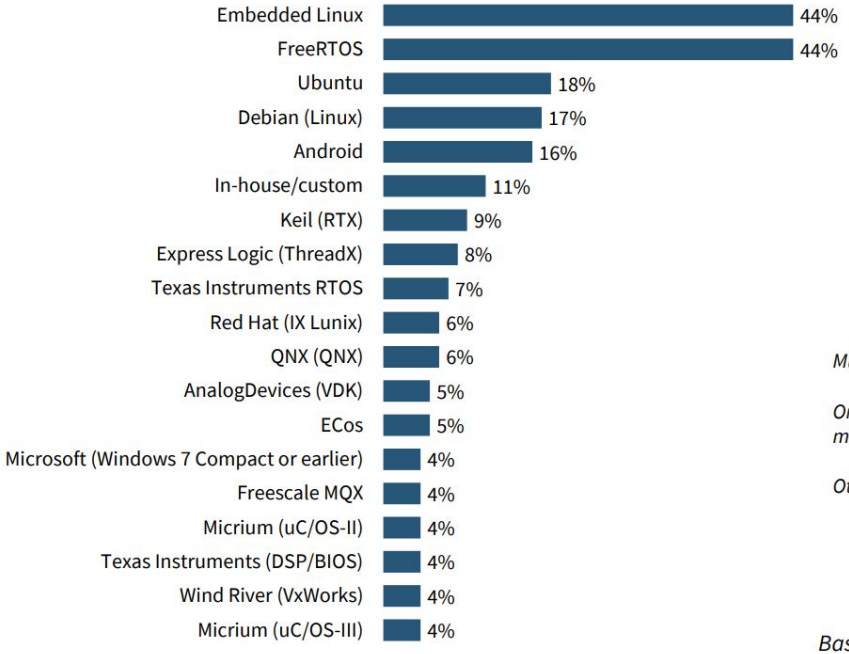


Total Respondents

Image source

Most popular embedded OSs – Embedded Linux, FreeRTOS and Ubuntu

Top 3 OSs are especially popular in APAC, while Embedded Linux is used more in the Americas



Multiple responses allowed

Only those with 4% or more total mentions shown

Other = 7%

Base = Those who will use an OS (566)

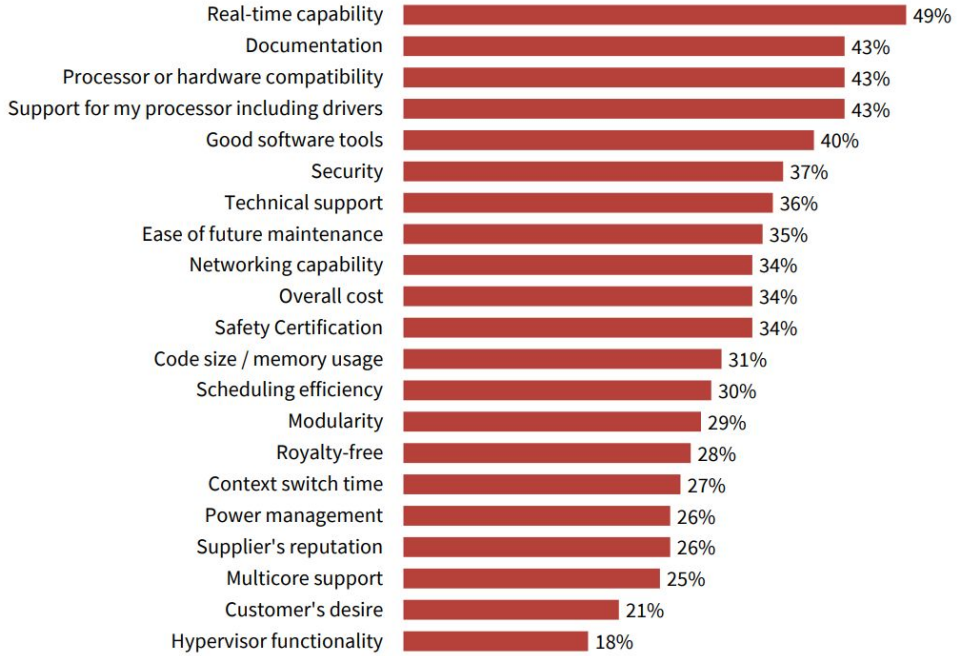
embedded survey

35. Please select the operating systems you are currently using or considering using in the next 12 months for a commercial product development project. ***(Only include non-RTOS operating systems that you embed into your projects.)***

Image source

Those using a commercial OS look for documentation, hardware compatibility and support to complement real-time capabilities

Large OEMs and APAC developers put particular emphasis on most commercial OS capabilities



'Very Important' Summary

Multiple responses allowed

Base = Those using commercial OS (200)

embedded survey

32. How important are the following factors regarding your decision to use a commercial operating system, RTOS, kernel, etc.?

Image source