[04] SCHEDULING

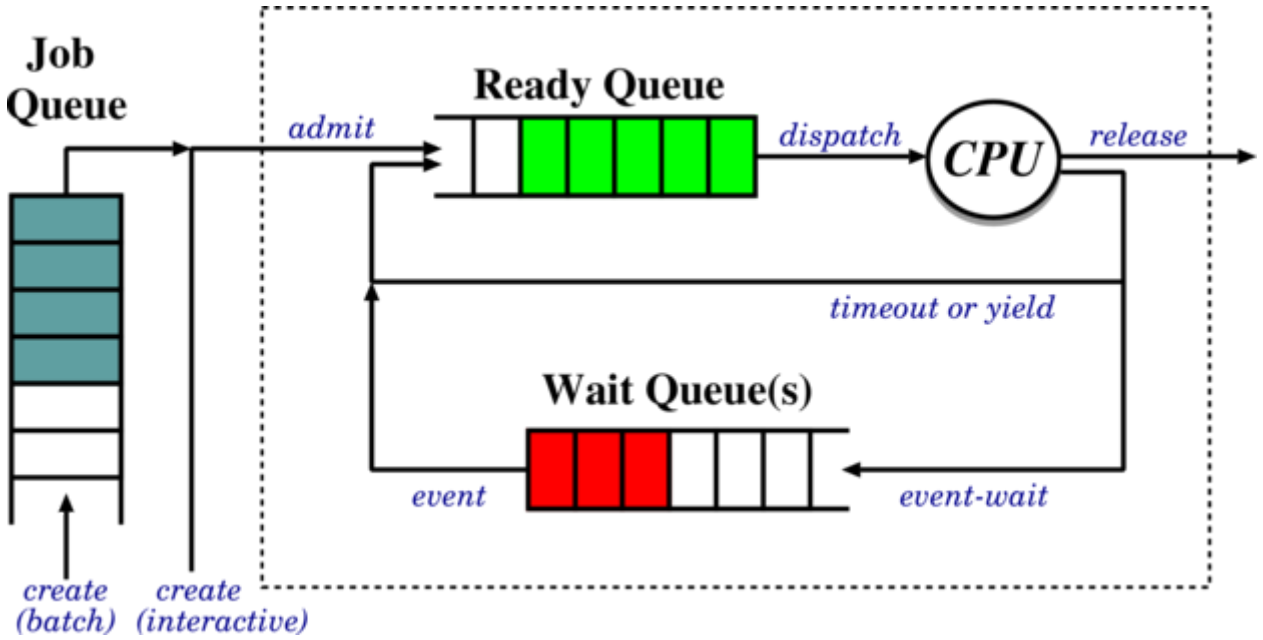
OUTLINE

- Scheduling Concepts
  - Queues
  - Non-preemptive vs Preemptive
  - Idling
- Scheduling Criteria
  - Utilisation
  - Throughput
  - Turnaround, Waiting, Response Times

SCHEDULING CONCEPTS

- **Scheduling Concepts**
  - **Queues**
  - **Non-preemptive vs Preemptive**
  - **Idling**
- Scheduling Criteria

QUEUES



- **Job Queue**: batch processes awaiting admission
- **Ready Queue**: processes in main memory, ready and waiting to execute
- **Wait Queue(s)**: set of processes waiting for an IO device (or for other processes)
  - **Job** scheduler selects processes to put onto the ready queue
  - **CPU** scheduler selects process to execute next and allocates CPU

PREEMPTIVE VS NON-PREEMPTIVE

OS needs to select a ready process and allocate it the CPU
When?

- ...a running process blocks (`running → blocked`)
- ...a process terminates (`running → exit`)

If scheduling decision is only taken under these conditions, the scheduler is said to be **non-preemptive**

- ...a timer expires (`running → ready`)
- ...a waiting process unblocks (`blocked → ready`)

Otherwise it is **preemptive**

NON-PREEMPTIVE

- Simple to implement:
  - No timers, process gets the CPU for as long as desired
- Open to *denial-of-service*:
  - Malicious or buggy process can refuse to yield

Typically includes an *explicit* `yield` system call or similar, plus *implicit* yields, e.g., performing IO, waiting

Examples: MS-DOS, Windows 3.11

PREEMPTIVE

- Solves denial-of-service:
  - OS can simply preempt long-running process
- More complex to implement:
  - Timer management, concurrency issues

Examples: Just about everything you can think of :)

IDLING

We will usually assume that there's always something ready to run. But what if there isn't?

This is quite an important question on modern machines where the common case is >50% idle

IDLING

Three options

1. Busy wait in scheduler, e.g., Windows 9x
    - Quick response time
    - Ugly, useless

IDLING

Three options

1. Busy wait in scheduler
2. Halt processor until interrupt arrives, e.g., modern OSs
    - Saves power (and reduces heat!)
    - Increases processor lifetime
    - Might take too long to stop and start

IDLING

Three options

1. Busy wait in scheduler
2. Halt processor until interrupt arrives
3. Invent an idle process, always available to run
    - Gives uniform structure
    - Could run housekeeping
    - Uses some memory
    - Might slow interrupt response

In general there is a trade-off between responsiveness and usefulness. Consider the important resources and the system complexity

SCHEDULING CRITERIA

- Scheduling Concepts
- **Scheduling Criteria**
  - **Utilisation**
  - **Throughput**
  - **Turnaround, Waiting, Response Times**

SCHEDULING CRITERIA

Typically one expects to have more than one option — more than one process is `runnable`

On what basis should the CPU scheduler make its decision?

Many different metrics may be used, exhibiting different trade-offs and leading to different operating regimes

CPU UTILISATION

> *Maximise the fraction of the time the CPU is actively being used*

Keep the (expensive?) machine as busy as possible

But may penalise processes that do a lot of IO as they appear to result in the CPU not being used

THROUGHPUT

Maximise the number of processes that complete their execution per time unit

Get useful work completed at the highest rate possible

But may penalise long-running processes as short-run processes will complete sooner and so are preferred

TURNAROUND TIME

Minimise the amount of time to execute a particular process

Ensures every processes complete in shortest time possible

WAITING TIME

> *Minimise the amount of time a process has been waiting in the `ready` queue*

Ensures an interactive system remains as responsive as possible

But may penalise IO heavy processes that spend a long time in the `wait` queue

RESPONSE TIME

> *Minimise the amount of time it takes from when a request was submitted until the first response is produced*

Found in time-sharing systems. Ensures system remains as responsive to clients as possible under load

But may penalise longer running sessions under heavy load

SUMMARY

- Scheduling Concepts
  - Queues
  - Non-preemptive vs Preemptive
  - Idling
- Scheduling Criteria
  - Utilisation
  - Throughput
  - Turnaround, Waiting, Response Times