428

Systems, Networks & Concurrency 2020

6

Scheduling

Uwe R. Zimmer - The Australian National University

429

Scheduling

References for this chapter

[Ben2006]
Ben-Ari, M
Principles of Concurrent and Distributed Programming
second edition, Prentice-Hall 2006

[AdaRM2012]
Ada Reference Manual - Language and Standard Libraries;
ISO/IEC 8652:201x (E)

[Stallings2001]
Stallings, William
Operating Systems
Prentice Hall, 2001

430

Scheduling

Motivation and definition of terms

Purpose of scheduling

431

Scheduling

Motivation and definition of terms

Purpose of scheduling

Two scenarios for scheduling algorithms:

1. Ordering resource assignments (CPU time, network access, …).
   ☞ live, on-line application of scheduling algorithms.
2. Predicting system behaviours under anticipated loads.
   ☞ simulated, off-line application of scheduling algorithms.

Predictions are used:

- *at compile time*: to confirm the feasibility of the system, or to predict resource needs, …
- *at run time*: to permit admittance of new requests or for load-balancing, …

432

Scheduling

Motivation and definition of terms

Criteria

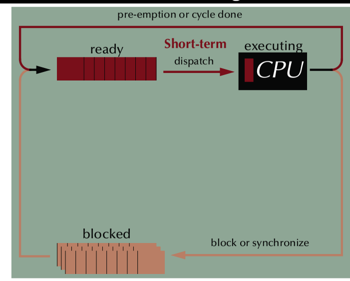
	Performance criteria:	**Predictability** criteria:
Process / user perspective:		
	minimize the …	minimize *deviation* from given …
Waiting time	minima / maxima / average / variance	value / minima / maxima
Response time	minima / maxima / average / variance	value / minima / maxima / deadlines
Turnaround time	minima / maxima / average / variance	value / minima / maxima / deadlines
System perspective:		
	maximize the …	
Throughput	minima / maxima / average	
Utilization	CPU busy time	

433

Scheduling

Definition of terms

Time scales of scheduling

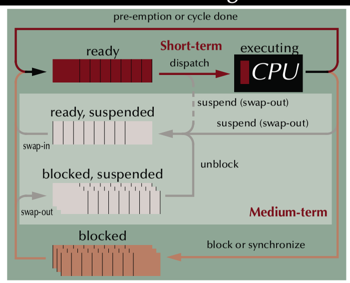


434

Scheduling

Definition of terms

Time scales of scheduling

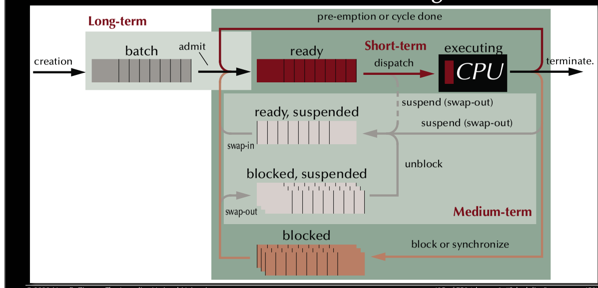


435

Scheduling

Definition of terms

Time scales of scheduling

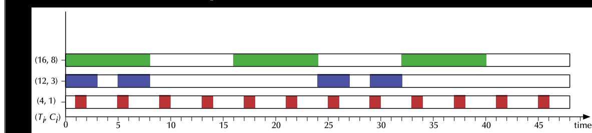


436

Scheduling

Performance scheduling

Requested resource times



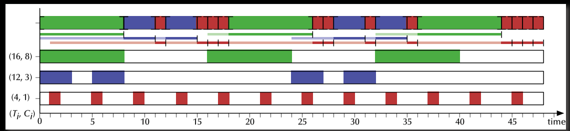
Tasks have an **average time between instantiations** of
and a constant **computation time** of

437

Scheduling

Performance scheduling

First come, first served (FCFS)



Waiting time: 0..11, average: 5.9 – **Turnaround time**: 3..12, average: 8.4

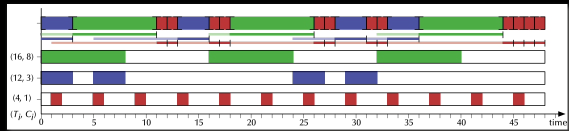
As tasks apply *concurrently* for resources, the actual sequence of arrival is non-deterministic.
☞ hence even a deterministic scheduling schema like FCFS can lead to different outcomes.

438

Scheduling

Performance scheduling

First come, first served (FCFS)



Waiting time: 0..11, average: 5.4 – **Turnaround time**: 3..12, average: 8.0

☞ In this example:
the average waiting times vary between 5.4 and 5.9
the average turnaround times vary between 8.0 and 8.4

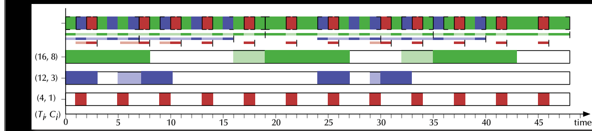
☞ **Shortest possible maximal turnaround time!**

439

Scheduling

Performance scheduling

Round Robin (RR)



Waiting time: 0..5, average: 1.2 – **Turnaround time**: 1..20, average: 5.8

☞ Optimized for swift initial responses.
☞ "Stretches out" long tasks.

☞ **Bound maximal waiting time!** (depended only on the number of tasks)

440

Scheduling

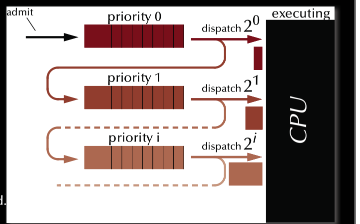
Performance scheduling

Feedback with 2^i pre-emption intervals

- Implement multiple hierarchical ready-queues.
- Fetch processes from the highest filled ready queue.
- Dispatch more CPU time for lower priorities (2^i units).

☞ Processes on lower ranks may suffer **starvation**.

☞ New and short tasks will be preferred.

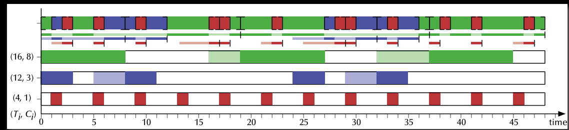


441

Scheduling

Performance scheduling

Feedback with 2^i pre-emption intervals - sequential



Waiting time: 0..5, average: 1.5 – **Turnaround time**: 1..21, average: 5.7

☞ Optimized for swift initial responses.
☞ Prefers short tasks and long tasks can suffer starvation.

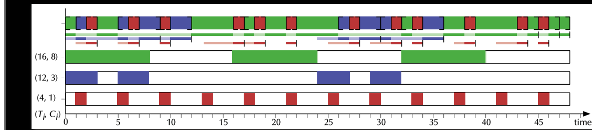
☞ **Very short initial response times!** and good average turnaround times.

442

Scheduling

Performance scheduling

Feedback with 2^i pre-emption intervals - overlapping



Waiting time: 0..3, average: 0.9 – **Turnaround time**: 1..45, average: 7.7

☞ Optimized for swift initial responses.
☞ Prefers short tasks and long tasks can suffer starvation.

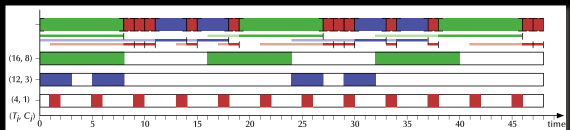
☞ **Long tasks are delayed until all queues run empty!**

443

Scheduling

Performance scheduling

Shortest job first



Waiting time: 0..11, average: 3.7 – **Turnaround time**: 1..14, average: 6.3

☞ Optimized for good average performance with minimal task-switches.
☞ Prefers short tasks but all tasks will be handled.

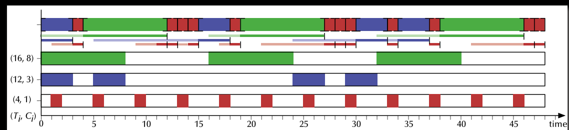
☞ **Good choice if computation times are known and task switches are expensive!**

444

Scheduling

Performance scheduling

Shortest job first



Waiting time: 0..10, average: 3.4 – **Turnaround time**: 1..14, average: 6.0

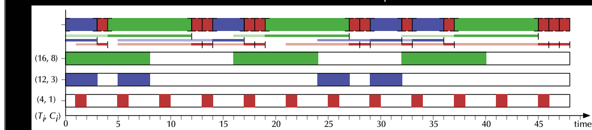
☞ Can be sensitive to non-deterministic arrival sequences.

445

Scheduling

Performance scheduling

Highest Response Ration $\frac{W_i + C_i}{C_i}$ First (HRRF)



Waiting time: 0..9, average: 4.1 – **Turnaround time**: 2..13, average: 6.6

☞ Blend between Shortest-Job-First and First-Come-First-Served.
☞ Prefers short tasks but long tasks gain preference over time.

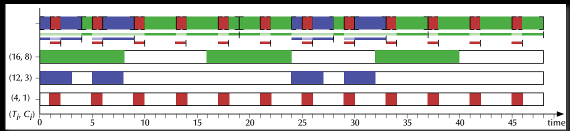
☞ **More task switches and worse averages than SJF but better upper bounds!**

446

Scheduling

Performance scheduling

Shortest Remaining Time First (SRTF)



Waiting time: 0..6, average: 0.7 – **Turnaround time:** 1..21, average: 4.4

- ☞ Optimized for good averages.
- ☞ Prefers short tasks and long tasks can suffer starvation..

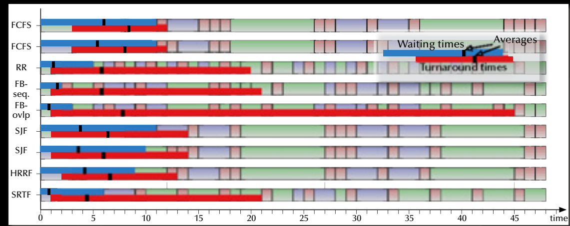
☞ **Better averages than Feedback scheduling but with longer absolute waiting times!**

447

Scheduling

Performance scheduling

Comparison (in order of appearance)

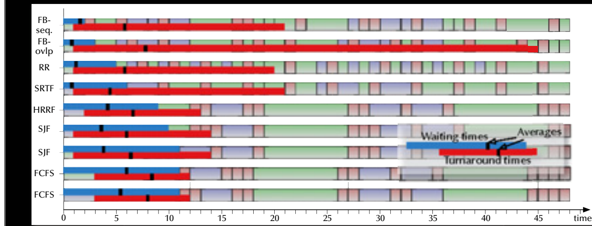


448

Scheduling

Performance scheduling

Comparison by shortest maximal waiting



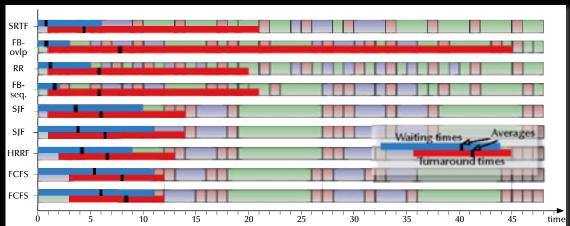
☞ Providing upper bounds to waiting times ☞ Swift response systems

449

Scheduling

Performance scheduling

Comparison by shortest average waiting



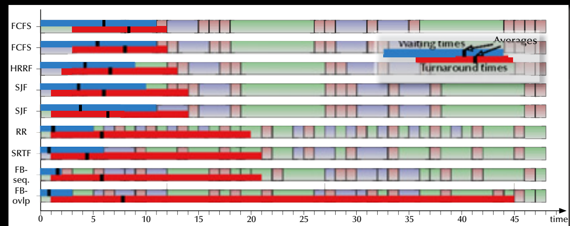
☞ Providing short average waiting times ☞ Very swift response in most cases

450

Scheduling

Performance scheduling

Comparison by shortest maximal turnaround



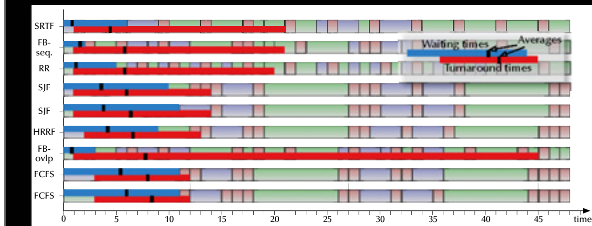
☞ Providing upper bounds to turnaround times ☞ No tasks are left behind

451

Scheduling

Performance scheduling

Comparison by shortest average turnaround



☞ Providing good average performance ☞ High throughput systems

452

Scheduling

Performance scheduling

Comparison overview

	Selection	Pre-emption	Waiting	Turnaround	Preferred jobs	Starvation possible?
Methods without any knowledge about the processes						
FCFS	$\max(W_i)$	no	long	long average & short maximum	equal	no
RR	equal share	yes	bound	good average & large maximum	short	no
FB	priority queues	yes	very short	short average & long maximum	short	no
Methods employing computation time C_i and elapsed time E_i						
SJF	$\min(C_i)$	no	medium	medium	short	yes
HRRF	$\max\left(\frac{W_i + C_i}{C_i}\right)$	no	controllable compromise	controllable compromise	controllable	no
SRTF	$\min(C_i - E_i)$	yes	very short	wide variance	short	yes

453

Scheduling

Predictable scheduling

Towards predictable scheduling …

Task requirements (Quality of service):

- ☞ Guarantee **data flow** levels
- ☞ Guarantee **reaction** times
- ☞ Guarantee **deadlines**
- ☞ Guarantee **delivery** times
- ☞ Provide **bounds** for the variations in results

Examples:

- Streaming media broadcasts, playing HD videos, live mixing audio/video, …
- Reacting to users, Reacting to alarm situations, …
- Delivering a signal to the physical world at the required time, …

454

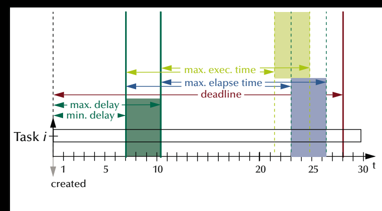
Scheduling

Predictable scheduling

Temporal scopes

Common attributes:

- Minimal & maximal **delay** after creation
- Maximal **elapsed time**
- Maximal **execution time**
- Absolute **deadline**



455

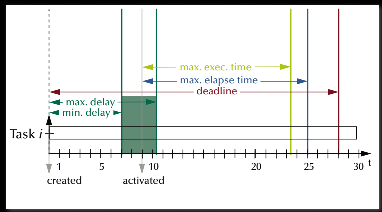
Scheduling

Predictable scheduling

Temporal scopes

Common attributes:

- Minimal & maximal **delay** after creation
- Maximal **elapsed time**
- Maximal **execution time**
- Absolute **deadline**



456

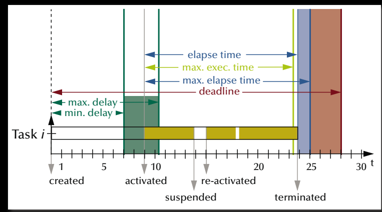
Scheduling

Predictable scheduling

Temporal scopes

Common attributes:

- Minimal & maximal **delay** after creation
- Maximal **elapsed time**
- Maximal **execution time**
- Absolute **deadline**



457

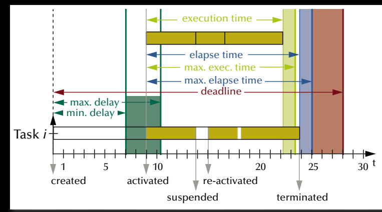
Scheduling

Predictable scheduling

Temporal scopes

Common attributes:

- Minimal & maximal **delay** after creation
- Maximal **elapsed time**
- Maximal **execution time**
- Absolute **deadline**



458

Scheduling

Predictable scheduling

Common temporal scope attributes

Temporal scopes can be:

Periodic	☞ controllers, routers, schedulers, streaming processes, …
Aperiodic	☞ periodic 'on average' tasks, i.e. regular but not rigidly timed, …
Sporadic / Transient	☞ user requests, alarms, I/O interaction, …

Deadlines can be:

Semantics defined by application

"Hard"	☞ single failure leads to severe malfunction and/or disaster
"Firm"	☞ results are meaningless after the deadline
	☞ only multiple or permanent failures lead to malfunction
"Soft"	☞ results are still useful after the deadline

459

Scheduling

Summary

Scheduling

- **Basic performance scheduling**
  - Motivation & Terms
  - Levels of knowledge / assumptions about the task set
  - Evaluation of performance and selection of appropriate methods
- **Towards predictable scheduling**
  - Motivation & Terms
  - Categories & Examples