

OPERATING SYSTEMS

ASSIGNMENT 4.1

HENRI HEYDEN, NIKE PULOW

stu240825, stu239549

Task 1

Algorithm	Pros	Cons
FCFS	Easy to implement	Convoy effect, really unfair
RR	Easy to implement	Still convoy effect, very much unfair for IO
SJF/SPN	No convoy effect	Unfair for slower processes, Starvation possible

Task 2

a) Multi-Level-Queues

b) Use different strategies for different processes, if we can group them according to their source or if they are mostly IO-processes or not. Also use strategies to decide the priorities of different kinds of processes groups.

c) A local scheduling strategies decides how to split up CPU-bursts of different processes inside a process group, a global scheduling strategy decides which processes group gets CPU-bursts.

d) Static global scheduling strategies give each process group a priority, which starves lower priority process groups if higher priority process groups exists, whereas dynamic global scheduling strategies switch between processes according to the time they use, which may for example be dividing the CPU usage time percentage each group may use.

Task 3

a) Edsger W. Dijkstra

b) The so-called “Dijkstra Algorithm”, which finds the shortest paths in a node graph.

Task 4

A binary semaphore is a counting semaphore which is initialized with the value 1 and does not exceed that value.

A binary semaphore represents a data source which may only be used by one process at a time while a counting semaphore can allow more processes to use a data source, or, for example, a code snippet.

Task 5

a) Active waiting is waiting for a conditional statement to be true, like for example a while loop.

Passive waiting is a process or thread blocking itself so that no CPU-resource is being wasted while waiting for a condition to be true, like in the active waiting example. This can be realized with calling `sleep()` or, like when waiting for a free resource, this can be realized by using semaphores.

b) Active waiting may waste CPU-bursts and processing power, since the condition is checked every time in every given CPU-burst. Also, active waiting processes most of the time are being given unnecessary CPU-bursts, since the OS may or may not even know that the process really is actively waiting.