SYSC4001 Assignment 3 Report
Section L3 - Group 3

Dante Gyetko (101296637)
Vlad Myrutenko (101301663)
Github Repo P1: https://github.com/DanteGyetko/SYSC4001_A3_P1
Github Repo P2: https://github.com/VladMyrutenko/SYSC4001_A3P2

21 input files were run through each of the three scheduling algorithms to produce 63 output files. The inputs were divided into three sections: 1 to 7 were CPU-bound (low I/O), 8 to 14 were balanced I/O and CPU, and 15 to 21 were I/O-bound.

CPU-bound:

Input File	Best Throughput	Best Mean TAT	Best Mean WT	Best Mean RT
1	Equal	Equal	Equal	Equal (no I/O)
2	EP_RR: 4/1150	EP: 536	EP: 205	EP: 86
3	EP_RR: 4/1100	EP_RR: 620	EP: 523	EP: 94
4	EP_RR: 4/2635	EP_RR: 1344	EP_RR: 637	EP: 493
5	RR: 3/639	RR: 406	RR: 181	RR: 89
6	RR: 4/1920	EP_RR: 942	EP_RR: 410	EP_RR: 257
7	RR: 4/3770	RR: 2310	RR: 1327	EP: 306

EP & EP_RR: By minimizing the context switching common in RR, they achieve better throughput and turnaround times for high-priority tasks.

Round Robin: Less effective due to the overhead of frequent context switching, which can slow down the completion of long processes.

EP and EP_RR are generally better for purely CPU-bound workloads.

Mixed:

Input File	Best Throughput	Best Mean TAT	Best Mean WT	Best Mean RT
8	EP: 3/1700	RR: 1533	RR: 566	EP: 286
9	EP_RR: 4/3980	RR: 2520	RR: 782	RR: 350
10	RR: 4/1550	RR: 1067	RR: 515	RR: 206
11	EP_RR: 4/765	EP_RR: 548	EP_RR: 177	EP: 101
12	EP: 4/1280	RR: 890	RR: 345	RR: 97
13	EP: 4/3335	EP_RR: 2315	EP_RR: 783	EP: 368
14	RR: 4/2580	EP_RR: 1687	EP_RR: 617	EP: 175

Round Robin: Its fairness ensures that short I/O-bound tasks are not starved by long CPU-bound tasks, leading to system responsiveness and lower average waiting times.

External Priorities: Performs poorly since a high-priority CPU-bound process can monopolize the CPU, causing long waits for everyone else and leaving I/O devices idle.

Round Robin is the most balanced and effective for mixed workloads.

I/O-bound:

Input File	Best Throughput	Best Mean TAT	Best Mean WT	Best Mean RT
15	RR: 4/805	EP_RR: 465	EP_RR: 115	EP_RR: 90
16	EP_RR, RR: 4/2530	RR: 1240	RR: 222	RR: 76
17	RR: 1666	EP_RR: 1144	EP_RR: 237	EP: 100
18	EP_RR: 4/1823	EP_RR, RR: 785	RR: 66	EP_RR: 52
19	EP_RR: 4/2465	EP_RR: 1045	EP_RR: 140	EP: 119
20	Equal	Equal	Equal	Equal
21	EP: 4/2028	RR: 1202	RR: 166	RR: 101

Input 20 was a strange case where only two processes were running. It worked out that there was no difference in metrics between each algorithm.

RR & EP_RR: Their preemptive nature allows the CPU to switch rapidly between tasks, ensuring processes can start their I/O operations quickly. This maximizes system throughput.

External Priorities: Its non-preemptive design leads to delays, as the CPU cannot switch to another process until the current one finishes its burst, even if short.

Round Robin and the EP_RR Hybrid are the clear winners for I/O-bound tasks.

Conclusion:

External Priorities: Effective for specific high-priority tasks but its non-preemptive nature and risk of starvation make it unsuitable for general purpose systems.

Round Robin: The most robust and fair algorithm. RR's preemptive behavior is ideal for systems with mixed or I/O-heavy workloads.

EP_RR: It has the effectives of EP for specific tasks with the introduction safety of Round Robin to mitigate starvation, making it versatile for general purpose systems.