**Lab 3**

**My code works as expected here are 2 screenshots**

Optimal Memory Allocation Policy for FCFS

A picture containing black, photo, sitting, screen

Description automatically generated

Paging (256) for SRTF

**A screenshot of a cell phone

Description automatically generated**

**Abstract:**

In an operating system, memory constantly needs to be allocated. In this lab I surveyed four different memory allocation. The four policies implemented are Optimal Memory Allocation Policy (OMAP), Paging, Best-Fit, and Worst-Fit. Metrics were gathered for each algorithm and compared. I will be comparing these using Average Turnaround Time, Average Response Time, Average Waiting Time, CPU Utilization, Throughput, and Average Time in Job Queue. The best policy turned out to be Optimal Memory Allocation Policy (OMAP). Best-Fit, Worst-Fit, and Paging at 256 bytes all did well and can be implemented in real world scenarios unlike OMAP.

**Introduction:**

The three CPU Scheduling Policies I implemented were First Come First Serve (FCFS), Shortest Remaining Time First (SRTF), and Round Robin (RR). The RR policy was implemented with several different quantum times (time that must elapse before the CPU is preempted. This was done to see how the difference in the quantum time would affect the times. Turnaround time is the time it takes for a process to finish, response time is when a process gets the CPU the first time, CPU busy time is the percentage of time a CPU is used by processes, throughput time is the number of processes completed divided by the observation time, waiting time is the total time spent by a process in the ready queue, and waiting time in the job queue is just the total time spent by a process in the job queue. I will discuss the specifics of the data next.

**Body:**

Below each plot graphically represents the relationship between different CPU scheduling policies and memory allocation strategies for a given metric.

Regarding the scheduling algorithm, the data shows that SRTF performs the lowest while FCFS performs the highest. With a large quantum RR behaves similarly to FCFS. This makes sense because non-preemptive tasks will generally have a higher turnaround time.

Regarding the memory policy, the data shows that this does not contribute to performance

Regarding the scheduling algorithm, the data shows that FCFS performed the worst and SRTF performed the best. Since AWT is dependent on time spent in the ready queue, it makes since that SRTF would be low because it selects the process that needs the least amount of time to complete.

Regarding the memory policy, the data shows that this does not contribute to performance

Regarding the scheduling algorithm, the data shows that round robin with a 10ms response time performed much better than any other algorithm which is to be expected. FCFS performed the worst which is expected. FCFS performed badly because it is non-preemptive.

Regarding the memory policy, the data shows that this does not contribute to performance

Regarding the scheduling algorithm, the data shows that the CPU was used the most for Round Robin with quantum of 500ms which makes sense because it does not switch tasks as much. FCFS also performed high which makes sense because it is non-preemptive.

Regarding the memory policy, the data shows that this does not contribute to performance

Regarding the scheduling algorithm, the data shows that FCFS performed the best while SRTF and round robin with low quantum performed the worst. It makes sense that round robin with quantum = 10ms would perform badly sense it is constantly switching tasks.

Regarding the memory policy, the data shows that this does not contribute to performance

The data above for ATAT, AWT, ART, CBT, and T are expected. Each of these metrics is mainly affected by the CPU scheduling algorithm. This was confirmed in lab 1. Despite different Memory Allocation Policies, there is not any significant change. The long-term scheduler and memory allocation policy are the only factors that affect the processes allocation of memory and admission from the job to the ready queue.

Regarding the scheduling algorithm, the data shows the Round Robin with a Q = 10ms performs the lowest while FCFS performs the highest. SRTF also performs very low and Round Robin with a Quantum = 500ms performs very high.

The data shows that Optimal Memory Allocation Policy performed the best for Average Waiting Time in Job Queue. This makes sense because OMAP is the theoretical limit. Best Fit and Worst Fit also performed well which makes since because both algorithms are contiguous. Paging with size 8192 performed the worst showing that increasing page size decreases AWTJQ. This is because the page size was too large causing low memory utilization.

**Conclusion:**

The data shows that the best memory allocation policy is OMAP. This algorithm provided the best results for the Average Waiting Time in Job Queue, which is the most important metric for memory allocation. The main problem with this algorithm is that it is a theoretical algorithm and cannot be implemented in a real-world scenario. Paging at 256 bytes, Best-Fit, and Worst-Fit all performed well and unlike OMAP can be implemented in the real world. The only algorithm I would not recommend is Paging at 8192 bytes because it performed much worse than any other algorithm. It is important to note that this lab was a simulation and not entirely representative of these algorithms’ real-world performance. Use in the real world may reveal one algorithm to be better in a specific situation.