Lab 1 Task

Abstract

The task is to implement the following scheduling algorithms and to compare the averages of various waiting times for the processes. In my case I have 2 average time for first execution,and the average time spent idle since entering the awaiting processes queue.

Implemented algorithms

1. First Come First Serve, aka. FIFO, aka FCFS Essentially this algorithm is a FIFO stack. We pop a process of the stack and after its finished we pop one again until the stack is emptied.
2. SJF, aka. “Smallest Job First”, Non-Preemptive In this algorithm we just substitute the FIFO stack for a Collection sorted by the job length and after a process is done executing we pop a new one off the collection.
3. SJF, aka. “Smallest Job First”, Preemptive Is exactly the same as algorithm no.2 but every set interval the currently executing process has a change of being downgraded to a PAUSE state if a shorter job appears in the Collection

Backend system

To ensure results consistent across algorithms a robust backend was required. To achieve that I have created a Simulation class responsible for the main execution “loop” as well as statistical data collection. An algorithm was just a plugin into the class. Again to ensure consistency I have created a type that the functions performing process selecting had to conform to.

```
SchedulerType: Type = Callable[[ProcessQueue, Process],
                               Optional[Process]]

def scheduler(queue: ProcessQueue,
              previous_process: Process) -> Optional[Process]:
    pass
```

Queue creation

To avoid the need for calculating a chance for a new process every “tick” and for keeping count of the globally created processes and to ensure consistent process creation across other algorithms. The queue is pre-allocated before starting the simulation. In addition to a unique name and execution length property each process had a time_of_arrival value which indicated after which tick number can the process be considered for execution.

The creation of the global queue is parametrized with 4 values:

1. Number of processes to create
2. Max time of arrival
3. Min execution length
4. Max execution length

Generic view of the algorithm function

Every "tick" of the execution loop the selected algorithm had 3 options: 1. To leave the current process for another "round" of execution 2. To swap the current process for a different one 3. To return None, which means that currently there are no processes in the queue