

FreeRTOS emulator for linux is used on my local machine. Below is a screenshot taken when running the executable.

```
Sending data...
Data sent!
Sending data...
matrix mult...
Data sent!
Sending data...
communication task period: 200ms
matrix task period: 575ms
Data sent!
Sending data...
Data sent!
Sending data...
matrix mult...
Data sent!
communication task period: 200ms
matrix task period: 559ms
Sending data...
Data sent!
Sending data...
Data sent!
matrix mult...
Sending data...
communication task period: 200ms
matrix task period: 547ms
Data sent!
Sending data...
Data sent!
Sending data...
Data sent!
matrix mult...
communication task period: 200ms
matrix task period: 573ms
Sending data...
Data sent!
Sending data...
Data sent!
Sending data...
Data sent!
matrix mult...
communication task period: 200ms
matrix task period: 575ms
Data sent!
Sending data...
Data sent!
Sending data...
Data sent!
communication task period: 200ms
matrix task period: 575ms
```

Why is "matrixtask" using most of the CPU utilization?

Due to a combination of time dependency of long task data/runtime and task's relative high priority, the matrixtask uses relatively high CPU utilization.

Why must the priority of "communicationtask" increase in order for it to work properly?

The priority must be increased in this case because the underlying RTOS uses priority based task scheduler to divide CPU time to multiple tasks.

What happens to the completion time of "matrixtask" when the priority of "communicationtask" is increased?

The completion time of matrixtask is increased since the scheduler allocates more CPU time to the communicationtask by giving up CPU time from matrixtask. The relative amount of increase depends on the tasks that have their priorities changed higher than the matrixtask's priority.

How many seconds is the period of "matrixtask"? (Hint: look at vApplicationTickHook() to measure it)

The measured period is around 0.570s.