

## Proposal: Priority Task Scheduler

### Description

The majority of Western—especially American—society struggles with time management. Students are especially prone to squandering large amounts of time procrastinating only to find themselves so packed next to the deadline that very little genuine learning occurs. Indeed, many individuals attempt to rectify their bad time habits through schedules that front-load work into their day which ends up wearing them out to the point that they no longer have the drive necessary to discipline themselves.

This web app is a software solution that helps others manage their time in bite-sized chunks at a time. Using techniques developed for CPUs and operating systems, this app will implement algorithms such as round-robin and feedback loops to give the end user an opportunity to make progress on multiple projects and assignments without overwhelming them.

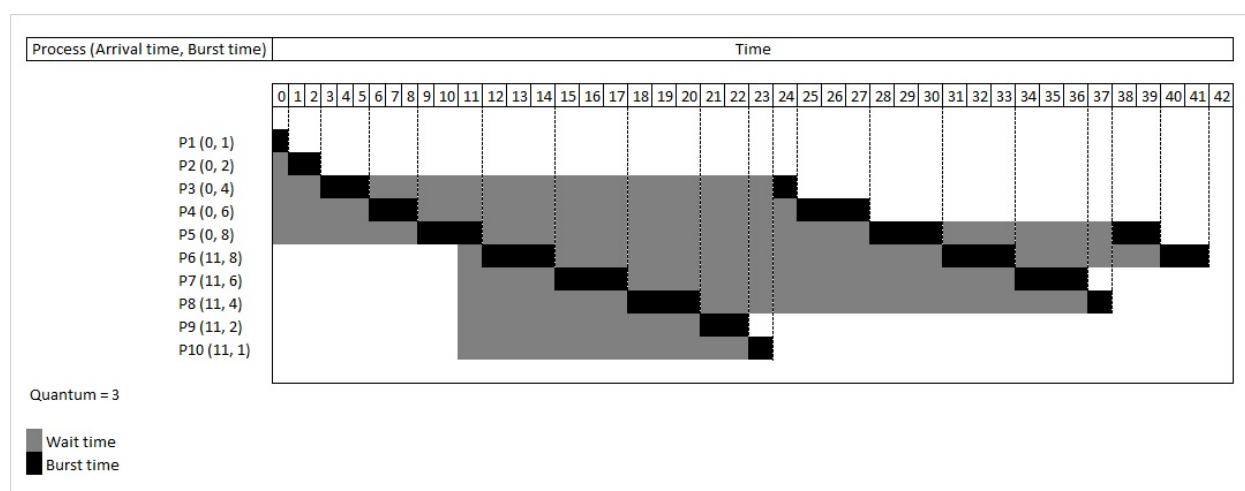


Figure 1. Example of a round-robin CPU scheduling graph

### Design

The app is relatively simple. The user structures his/her schedule for the day, week, etc. and prioritizes tasks to be done. The user, once ready, then starts the schedule timer which cycles through tasks to start, one by one, pausing each at regular intervals (as determined by the user) to take a break and begin or resume another task. The user can specify what constitutes valid “breakpoints” in each task so as to achieve reasonable milestones that nonetheless boost their confidence in their ability to finish their workload on time.

## Database Structure

The database will not be complicated as there will not be many different types of values to store. Virtually all data will revolve around tasks in a somewhat hierarchical manner. A given project, for example, would look something like:

Key	Value
Name	Project Proposal
Course	CS 313
Milestones	<table>

Table 1. Base key-value pairing for a “project” object

In this example, the “Milestones” key would have a reference value pointing to the following:

Task Description	Time to Completion	Required
Design the layout	10 minutes	Yes
Fill in the first two paragraphs	15 minutes	Yes
Fill in the first database paragraph	15 minutes	Yes
Fill in the last two paragraphs	20 minutes	Yes
Add images and drawings	20 minutes	No

Table 2. Data pairing for a “milestones” reference table

## Database Retrieval & Update

The user is initially presented with an empty UI—most likely a grid or a fillable list—to enter a new project in. Each new blank form constitutes a new project to create. The user can also update the projects by selecting an existing project field from a list or grid which then retrieves the data from the database and displays it in a new dialog. The user also has preferences that can be set, changed and saved to the database such as: whether tasks can be preempted; the maximum time a task can be scheduled for; etc. Other UI/presentation options will be configurable via this settings/preferences dialog.

Each time a new cycle of the schedule timer begins, its length is determined by the “Time to Completion” field of the next task; once that task’s timer runs out, a new task from a new project/assignment is called in and begins. This continues until all tasks for each project given by the user are completed, or, if the user continues adding tasks, it can run indefinitely. The user can also pause the timer at any time, as necessary.

The server will send queries through a base PHP file that will read from and update local JSON, XML or CSV files. Checks will need to be placed at regular intervals including hashes, checksums, etc. so as to ensure that no data was lost during execution, retrieval or posting. Data redundancy and local storage copies may also be required to ensure that virtually no data is ever lost.