

Plan

TABLE OF CONTENTS

- 1 [Checkpoint 1](#)
 - 2 [Checkpoint 2](#)
 - 3 [Final](#)
-

We provide you a suggested order of implementation as well as the specifications for each checkpoint based on our and past students' experiences. However, this is merely a suggestion and you may elect to approach the project entirely differently.

Keep in mind that checkpoints are not graded.

Checkpoint 1

Start the project with implementing the efficient alarm clock. This is the simplest task within the project, so try not to spend too much time on it.

Checkpoint 2

Implement your strict priority scheduler.

Final

Finish the project by implementing user threads. If you get stuck on `multi-oom` or `exit-clean` tests, we advise that you skip those and come back at the end.

You should be able to implement user threads independently of the efficient alarm clock and strict priority scheduler. The alarm clock does not have an exposed interface via system calls, so the efficient alarm clock and user threads are completely independent. There is some overlap between the strict priority scheduler and user threads because user threads use both the scheduler and locks.

However, the tasks should still be fairly independent of one another, since all user threads should have the same priority (`PRI_DEFAULT`).
