Jacob Francis

Worksheet 1.3: OS Basics, Timing Diagrams

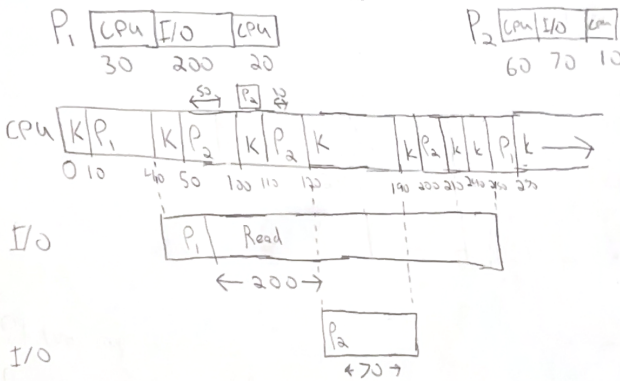
Q1. An OS running on a **single CPU** has the following two processes:

- P_1 consists of a 30-ms CPU burst followed by a 200-ms I/O request followed by a 20-ms CPU burst
- P_2 consists of a 60-ms CPU burst followed by a 70-ms I/O request followed by a 10-ms CPU burst.

Draw a complete timing diagram showing how a time-sharing operating system will handle these processes under the following assumptions:

- Each I/O request goes to a **different** device.
- The kernel starts each I/O request right at the beginning of the kernel time.
- The I/O device will start processing a queued request as soon as the current request is done (immediately if it is not currently processing another request and the queue is empty).
- Whenever P_1 and P_2 are **both** ready, the scheduler will give the CPU to **P_1**.
- The time quantum is 50 ms.
- The kernel time (including initiation of I/O requests, scheduling and context switching) is 10 ms.
- Ignore any delay or transition time that was ignored in class.

Show the CPU and the I/O devices in your diagram and align the CPU time with the I/O time.



Q2. An OS running on a **single CPU** has the following two processes:

- P_1 consists of a 30-ms CPU burst followed by an I/O request followed by a 40-ms CPU burst
- P_2 consists of a 60-ms CPU burst followed by an I/O request followed by a 20-ms CPU burst

Draw a complete timing diagram showing how a time-sharing operating system will handle these processes under the following assumptions:

- Each I/O request goes to a different device, and each request takes 80 ms.
- The I/O device starts processing the next request as soon as the current request is done (immediately if it is not currently processing another request and the queue is empty).
- Whenever P_1 and P_2 are **both** ready, the scheduler will give the CPU to the process that has been sitting in the ready queue for a longer period of time, that is, the ready queue is a First-In-First-Out (FIFO) queue. When the kernel handles an I/O completion interrupt, it will enter the interrupted process into the ready queue **before** entering the process whose I/O request has been completed. For the very first selection, assume that the scheduler will select P_2, that is, P_2 will get the CPU at time 10.
- The time quantum is 40 ms.
- The kernel time (including initiation of I/O requests, scheduling and context switching) is 10 ms.
- The OS starts each I/O request right at the beginning of the kernel time.
- Ignore any delay or transition time that was ignored in class.

Show both the CPU and the I/O device in your diagram and align the CPU time with the I/O time.