Systems Programming (2024 Fall)
Programming Assignment 1

B12902110 呂承諺

1. (a) Busy waiting means a process is repeatedly testing if a condition has become true. In this assignment, that is testing if data is available to be read.

   (b) In this assignment, we can avoid busy waiting by using `select()` or `poll()` with a properly configured timeout. This way, the server only continues execution when there is data is ready to be read.

   (c) It is still possible to have busy waiting even with `select()` or `poll()`, if the timeout is set to 0.

2. (a) Starvation refers to an I/O request not getting data, resulting in an indefinite wait.

   (b) In this assignment, it is possible for a request to encounter starvation if our code handles a request using infinite loops. To avoid this, we use a state-based approach, keeping track of every request's state and changing its state accordingly after each command. This way, we only have to handle commands when input data is available, thus preventing indefinite waits.

3. Since a process could never test for its own lock via `fcntl()`, we need to keep track of which seats are reserved or paid by which requests handled by this server process. This can be done by creating an array of seat statuses for each shift, maintaining the request's ID that has reserved or paid for the shift.

4. We use advisory record locking provided by `fcntl()`. Before writing a byte range (a seat is essentially a byte) in a shift's record file, we first need to check is another process is holding a lock to that byte range. If that is true, we have to ensure that our process will not write to that byte range.