

令和7年度東京大学大学院情報理工学系研究科修士課程・博士後期課程入学試験
筆記試験における出題ミスについて

東京大学大学院情報理工学系研究科

このたび、令和7年度大学院情報理工学系研究科入学試験（一般選抜及び社会人特別選抜、夏入試）において、コンピュータ科学専攻筆記試験（専門科目）の試験問題の一部に出題ミスがあることが判明いたしました。出題ミスの内容及び本研究科の対応は次のとおりです。

今後、再びこのような事態が起こらぬよう必要な対策を講じ、再発防止に取り組んでまいります。

1 出題ミスの内容

専門科目の問題3（別紙参照）中の(c)のケースについて、小問(1)(3)(4)に意図せず2通りの正解があった。

なお、小問(2)については、小問(1)に連動して2通りの計算過程が存在するものの、最終的な正解は同一である。

2 出題ミスへの対応

問題3(c)のケースでは、2通りのいずれの解答も正解とし、それ以外の解答は不正解として採点を行った。この措置により、入学試験の合否判定において不利益を被る受験者がいないよう対応した。

問題 3

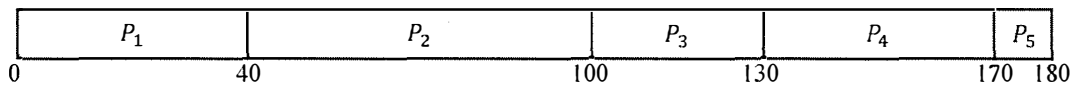
オペレーティングシステムにおける、プロセス P_i ($i = 1, 2, 3, 4, 5$) のスケジューリングを考える。 P_1, P_2, P_3, P_4, P_5 の到着時刻はそれぞれ 0, 5, 10, 15, 20, 処理時間は 40, 60, 30, 40, 10 である。次の 4 つのスケジューリングアルゴリズムを考える。(a) first come, first serve (FCFS), (b) (non-preemptive) shortest job first (SJF), (c) shortest remaining time first (SRTF), (d) round robin (RR) (time quantum は 30)。ただし、一度に実行できるプロセスは一つだけで、言及のないコストは無視できるものとする。

また、バイナリセマフォ S_j ($j = 1, 2$) を取得、解放するための操作をそれぞれ $wait(S_j)$, $signal(S_j)$ とする。操作 $wait(S_j)$ は、 S_j を取得できなかったら、実行したプロセスを S_j の待ちキューの末尾に入れる。操作 $signal(S_j)$ は、 S_j を解放したあと、 S_j の待ちキューが空でなければ、その先頭のプロセスを実行可能キューに移動する。

なお、プロセスの待ち時間は、実行可能キューまたは S_j ($j = 1, 2$) の待ちキューで待つ時間の合計とする。プロセスの平均待ち時間とは、各プロセス P_i ($i = 1, 2, 3, 4, 5$) の待ち時間の平均のことである。

(a)~(d) のそれぞれのスケジューリングアルゴリズムについて、以下の問いに答えよ。ただし、以下の問い (1), (2) では、プロセスはセマフォ操作はおこなわないものとする。

- (1) P_i ($i = 1, 2, 3, 4, 5$) がスケジュールされる時刻を、以下のようなガントチャートの形で描け。



- (2) プロセスの平均待ち時間を求めよ。
- (3) P_1, P_3, P_5 は起動時に $wait(S_1)$, 終了時に $signal(S_1)$ をそれぞれ実行し、 P_2, P_4 はセマフォ操作をおこなわないとする。このときのプロセスの平均待ち時間を求めよ。
- (4) P_i が以下のようにセマフォ操作をおこなう場合を考える。
- P_1 と P_2 は、起動時に $wait(S_1)$, 実行時間が 15 経過後に $wait(S_2)$, 終了時に $signal(S_2)$ 及び $signal(S_1)$ をこの順に実行する。
 - P_3 と P_4 は、起動時に $wait(S_2)$, 実行時間が 15 経過後に $wait(S_1)$, 終了時に $signal(S_1)$ 及び $signal(S_2)$ をこの順に実行する。
 - P_5 はセマフォ操作をおこなわない。

このとき、デッドロックが発生する場合はその発生時刻を求めよ。デッドロックが発生しない場合はプロセスの平均待ち時間を求めよ。

Problem 3

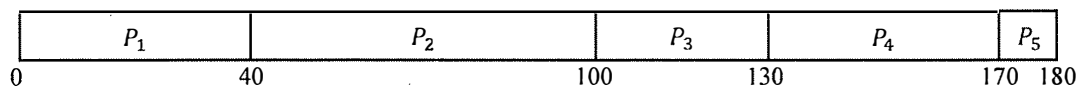
Consider the scheduling of processes P_i ($i = 1, 2, 3, 4, 5$) in an operating system. The arrival times of P_1, P_2, P_3, P_4 and P_5 are 0, 5, 10, 15 and 20, and processing times are 40, 60, 30, 40 and 10, respectively. We consider the following four scheduling algorithms: (a) first come, first serve (FCFS), (b) (non-preemptive) shortest job first (SJF), (c) shortest remaining time first (SRTF), and (d) round robin (RR) (time quantum is 30). Here, we assume that only one process can run at a time and unmentioned costs can be ignored.

Also, the operations to acquire and release binary semaphores S_j ($j = 1, 2$) are denoted as $wait(S_j)$ and $signal(S_j)$, respectively. If the operation $wait(S_j)$ cannot acquire S_j , it puts the executing process at the end of the waiting queue for S_j . The operation $signal(S_j)$ releases S_j , and if the waiting queue for S_j is non-empty, then the first process in the waiting queue is moved to the ready queue.

Here, the waiting time of a process is the total amount of time the process spends waiting in the ready queue or the waiting queue for S_j ($j = 1, 2$). The average waiting time of the processes is the average of the waiting time of each process P_i ($i = 1, 2, 3, 4, 5$).

For each of the scheduling algorithms (a) to (d), answer the following questions. In questions (1) and (2), assume that the processes do not perform any semaphore operations.

- (1) Draw the time P_i ($i = 1, 2, 3, 4, 5$) is scheduled in the form of a Gantt chart as shown below.



- (2) Calculate the average waiting time of the processes.
- (3) Suppose that P_1, P_3 , and P_5 perform $wait(S_1)$ at startup and $signal(S_1)$ at termination, respectively, and P_2 and P_4 do not perform any semaphore operations. Calculate the average waiting time of the processes in this case.
- (4) Consider the case where P_i performs semaphore operations as follows.
- P_1 and P_2 perform $wait(S_1)$ at startup, $wait(S_2)$ after a runtime of 15 elapsed, and $signal(S_2)$ and $signal(S_1)$ at termination in this order.
 - P_3 and P_4 perform $wait(S_2)$ at startup, $wait(S_1)$ after a runtime of 15 elapsed, and $signal(S_1)$ and $signal(S_2)$ at termination in this order.
 - P_5 does not perform any semaphore operations.

In this case, if a deadlock occurs, find the time when the deadlock occurs. If no deadlock occurs, calculate the average waiting time of the processes.