

ANKARA UNIVERSITY - DEPARTMENT OF COMPUTER ENGINEERING
COM 334/BLM 334 Operating Systems/İşletim Sistemleri - Final
9 June 2020 - Duration 150 minutes (Süre 150 dk)

Warning: You have upload your solutions untill 16:00. Write your solutions on paper, scan (or photo capture through a mobile application such as CamScanner, clearscanner etc.) and submit them as a single .pdf file. Your solutions have to be handwritten. Good luck. **TR:** Çözümlerinizi 16: 00'a kadar yüklemeniz gerekiyor. Çözümlerinizi kağıda yazın, tarayın (veya CamScanner, clearscanner gibi bir mobil uygulama aracılığıyla fotoğraf çekimi) ve bunları tek bir .pdf dosyası olarak gönderin. Çözümlerinizi el yazısı ile yazılmalıdır. İyi şanslar.

- 1** (%26) Consider the following set of jobs to be scheduled for execution on a single CPU system
- Draw a Gantt chart showing **preemptive SRJF scheduling**.
 - Calculate the average waiting time and average turnaround time.
 - Now assume that, process P5 used the CPU for (5ms, 3ms, 5ms) in its 3 previous executions. What will be the next CPU burst prediction for P5 if the system uses **exponential averaging** with equal weights for most recent past and for the earlier cpu usages.
- (TR: Tek bir CPU sistemi üzerinde yürütme için zamanlanacak işlemler kümesi veriliyor.**
- Kesilebilen SRJF algoritması için zaman yerleşimini gösterin.
 - Ortalama bekleme süresi ve ortalama tamamlama süresini hesaplayınız.
 - Şimdi, P5 işleminin önceki 3 yürütmesinde CPU' yu (5ms, 3ms, 5ms) kullandığını varsayın. Eğer sistem en yakın geçmiş ve ondan daha öncekiler için eşit ağırlıklarla üstel ortalama yöntemini kullanıyorsa P5 için bir sonraki CPU kullanımı tahmini ne olur?

Process	Arrival time	CPU time
P1	0	4
P2	4	6
P3	5	1
P4	7	3
P5	12	5

- 2.** (%25) A database is to be shared among several concurrent processes. Some of these processes may want only to read the database (readers), whereas others may want to update (that is, to read and write) the database (writers). **Design an algorithm** in pseudocode to guarantee such that: no data inconsistency will occur, the readers are able to read simultaneously, **all the writers are able to write before readers**. **(TR: Bir veritabanı aynı anda birkaç işlem arasında paylaşılacaktır. Bu işlemlerin bazıları yalnızca veritabanını (okuyucular) okumak isteyebilir, diğerleri ise veritabanını (yazarlar) güncellemek (yani okumak ve yazmak) isteyebilir. Aşağıdakileri garanti etmek için sözde kodda bir algoritma tasarlayın: veri tutarlılığı sorunu olmamalı, okuyucular aynı anda okuyabilmeli, tüm yazarlar okuyuculardan önce yazabilmelidir.)**

- 3.** (%25) Design a **Resource Request** Procedure which uses the deadlock avoidance algorithm. Your procedure in pseudo code should include the necessary steps for realising the deadlock avoidance method. **(TR: Bir ölümcül kilitlenme önleme algoritması kullanan bir Kaynak Talep Prosedürü tasarlayınız. Sözde kodda yazılmış Kaynak Talep prosedürünüz bir kilitlenme önleme yöntemi için gerekli adımları içermelidir.**

- 4.** (%24) For a virtual memory system, show the sequence of frames put into main memory with following reference string. Take the number of empty frames as $f = ((\text{last_digit_of_your_student_id}) \bmod 4) + 3$.

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2

- Using 'f' available frames and the OPT algorithm.
- Using 'f' available frames and LRU.

(TR: Bir sanal hafıza sisteminde verilen dizi için a) f tane boş frame varken OPT ve b) f tane boş frame varken LRU. Boş frame sayısını $f = ((\text{öğrenci_numaranızın_son_rakamı}) \bmod 4) + 3$ olarak alınız?