CSIT 345 Summer 2021 Homework 4

1. What are the four necessary conditions for deadlock? Give some brief description. ( 8 pts )

Mutual exclusion – the resource is unshareable.

Hold and wait or partial allocation – the process holds the resources while waiting for other processes.

No pre-emption – the process with the resources cant have them taken so any other processes waiting for the resource are in deadlock.

Circular wait – every process holds the resource of the next process in the chain

1. In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes come and go, new resources are bought and added to the system. If deadlock is controlled by the banker’s algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances? (15 pts)

a. Increase Available (new resources added)

b. Decrease Available (resource permanently removed from system)

c. Increase Max for one process (the process needs or wants more

resources than allowed).

d. Decrease Max for one process (the process decides it does not need

that many resources)

e. Increase the number of processes

f. Decrease the number of processes

Increase available – i.e. increasing the number of resources available for the required program can never cause an issue if it already is in the safe state.

Decrease available – this can cause issue and a create a lock state.

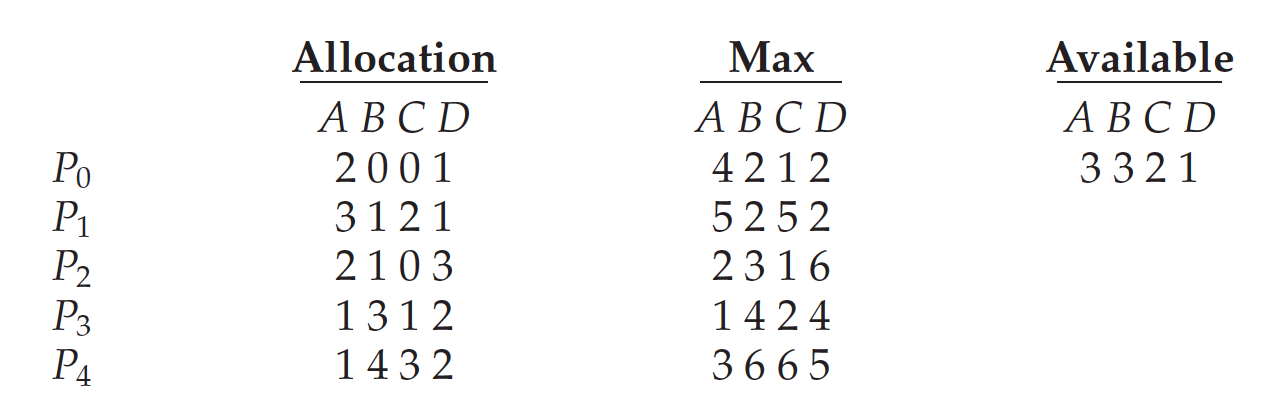
Increase max – can lead to a deadlock because of a new need

Decrease max – can’t cause a dead lock

Increasing number of processes- no deadlock unless resources were not allocated properly

Decreasing number of processes- can be changed with no problem.

1. Consider the following snapshot of a system: (12 pts)



Answer the following questions using the banker’s algorithm:

1. Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.

A B C D A B C D A B C D A B C D

P1 2 0 0 1 4 2 1 2 2 2 1 1 3 3 2 1 P2 3 1 2 1 5 2 5 2 2 1 3 1 5 3 2 2

3 3 2 1 + 2 0 0 1 = 5 3 2 2 5 3 2 2 + 3 1 2 1 = 8 4 4 3

P3 2 1 0 3 2 3 1 6 0 2 1 3 8 4 4 3 P4 1 3 1 2 1 4 2 4 0 1 1 2 10 5 4 6

8 4 4 3 + 2 1 0 3 =10 5 4 6 10 5 4 6 + 1 3 1 2 = 11 8 5 8

P5 1 4 3 2 3 6 6 5 2 2 3 3 11 8 5 8

11 8 5 8 + 1 4 3 2 =12 12 8 1 0

1. If a request from process P1 arrives for (1, 1, 0, 0), can the request be granted immediately?

Needs 2211 and it requests 1100 so the request is lower than the need therefore it would be granted

1. If a request from process P4 arrives for (0, 0, 2, 0), can the request be granted immediately?

Needs 0112 and it requests 0020 so the request is lower than the need therefore it would be granted

1. Explain the difference between internal and external fragmentation. ( 10 pts)

Internal fragmentation – a fixed sized partition which is assigned to a program or file

External fragmentation – all assigned blocks are moved to one side, so that contiguous space is gained.

1. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375KB (in order)? Compare these three algorithms in terms of how efficiently they use memory. (15pts)

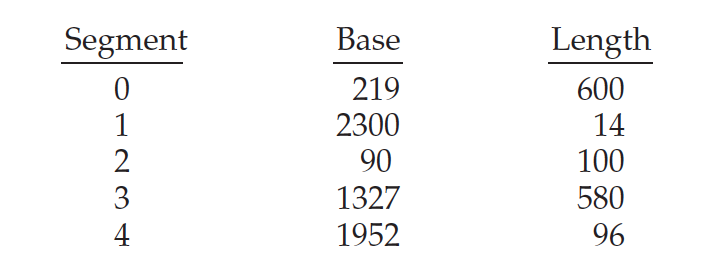
First fit - 115 KB is put in 300 KB partition, 500 KB is put in 600 KB partition, 358 KB is put in 750 KB partition, 200 KB is put in 350 KB partition, 375 KB is put in 392 KB partition

Best fit- 115 KB is put in 125 KB partition, 500 KB is put in 600 KB partition, 358 KB is put in 750 KB partition, 200 KB is put in 200 KB partition, 375 KB is put in 392 KB partition

Worst fit - 115 KB is put in 750 KB partition, 500 KB is put in 635 KB partition, 358 KB is put in 600 KB partition, 200 KB is put in 350 KB partition, 375 KB has to wait

Best fit is most efficient.

1. Consider the following segment table: (10pts)



What are the physical addresses for the following logical addresses?

a. 0,430

b. 1,10

c. 2,500

d. 3,400

e. 4,112

a) 219 + 430 = 649

b) 2300+ 10 = 2310

c) 90 +500 = 590

d) 1327 + 400 = 1727

e) 1952 + 112 = 2064

1. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers): (10pts)

a. 3085

b. 42095

c. 215201

d. 650000

e. 2000001

a) 3085 = 0000 1100 0000 11012 Page number = 00000000000000001010 01 Offset = 0000001101

Page number = 3 Offset = 13

b) 42095 = 1010 0100 0110 11112 Page number = 0001101111 Offset = 0001101111

Page number = 41 Offset = 111

c) 215201=11 0100 1000 1010 0001Page number = 0000000000000011010010 Offset = 0010100001

Page number = 210 Offset = 161

d) 650000=1001 1110 1011 0001 0000 Page number = 0000000000001001111010 Offset = 1100010000

Page number = 634 Offset = 784

e) 2000001=1010 0100 0110 1111 Page number = 0000000000000000101001 Offset = 0001101111

Page number = 41 Offset = 111

1. Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following? (10pts)

a. A conventional single-level page table

b. An inverted page table

1. 2^20
2. 2^17
3. Consider a paging system with the page table stored in memory. (10pts)

a. If a memory reference for either page table access or word in memory access takes 200 nanoseconds, how long does a paged memory reference take without TLB?

b. If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time?

1. 0 + 1 \*2 \* 200ns = 400 ns
2. 0.75 \* (0 + 200) + 0.25 \* (0 + 2 \* 200) = 250 ns