CPSC 457 Assignment4_ Muzhou,Zhai_10106810_T02

Q1 – Written question (5 marks)

Provide a simple example of a system in an unsafe state where the processes could finish without entering a deadlock. Show the state, and two sequences of execution: one leading to a deadlock and the other leading to the completion of all processes. Hint: you should not need more than 3 processes and one resource type (with multiple instances).

Sequence 1 (Currently only 1 resource available):
an unsafe state that the processes finish with entering a deadlock.

	Allocation	Maximum	Needed
P0	1	4	3
P1	1	6	5

-Both process 0 and process 1 can not be completed with the 1 available resource, this is an unsafe state. Both of them waiting for resource release, so that these processes go to deadlock.

Sequence 2 (Currently only 1 resource available):
an unsafe state that the processes finish without entering a deadlock.

	Allocation	Maximum	Needed
P0	3	4	1
P1	4	10	6
P2	3	8	5

-This is an unsafe state because only P0 can be completed, P1 and P2's completeness cannot be garentted.
-Firstly, with the 1 available resource, Process 0 can be completed. Then 4 resources can be released.
-Secondly, Process 2 can release 1 resource so that there are totally 5 resources, and P2 can be completed. Then 8 resources can be released.
-Finally P1 can be completed with 6 resources.

Q2 – Written question (5 marks)

What is the smallest value of 'x' that can keep the system in a safe state? Once you find the 'x' value, find a safe execution order using the Safety Algorithm. Include a step-by-step walk-through of the Safety Algorithm.

X=1 and we get the table as follow:

	Allocation	Maximun	Needed	Available
P0	10211	11213	01002	00112
P1	20110	22210	02100	
P2	11010	21310	10300	
P3	11110	11221	00111	

P3: Available 00112 + Allocation 11110 = 11222
P0: Available 11222 + Allocation 10211 = 21433
P2: Available 21433 + Allocation 11010 = 32443
P1: Available 32443 + Allocation 20110 = 52553
So that this state is safe, the execution sequence is P3, P0, P2, P1.

Q3 – Written question (5 marks)
Assume an OS has five free memory partitions of 100KB, 500KB, 200KB, 300KB and 600KB:

free 100KB	P10 30KB	free 500KB	P11 30KB	free 200KB	P12 30KB	free 300KB	P13 30KB	free 600KB

The OS needs to place 4 new processes in memory in the following order: P1 of 212KB, P2 of 417KB, P3 of 112KB and P4 of 426KB. Draw the diagrams of the partitions after the OS has placed the processes using 4 different algorithms: first-fit, best-fit, worst-fit and next fit. The resulting diagrams must show the size of each partition, and the status of each partition. If a process cannot be placed, indicate that below the diagram. Please start the placement algorithms from the first partition.

-**First fit algorithm:** find the first hole that is big enough, leftover space becomes new hole.
(unit: KB)

free 100	P10 30	P1 212	P3 112	free 176	P11 30	free 200	P12 30	free 300	P13 30	P2 417	free 183

P4 cannot be placed.

-**Best fit algorithm**: find the smallest hole that is big enough, leftover space becomes new hole (but very likely useless)
(unit: KB)

free 100	P10 30	P2 417	free 83	P11 30	P3 112	free 88	P12 30	P1 212	free 88	P13 30	P4 426	free 74	

-**Worst fit algorithm**: find the largest hole, leftover space is likely to be usable
(unit: KB)

free 100	P10 30	P2 417	free 83	P11 30	free 200	P12 30	free 300	P13 30	P1 212	P3 112	free 276

P4 cannot be placed.

-**Next fit algorithm**: same as first fit, but start searching at the location of last placement
(unit: KB)

free 100	P10 30	P1 212	free 288	P11 30	free 200	P12 30	free 300	P13 30	P2 417	P3 112	free 71

P4 cannot be placed.

Q4 – Written question (5 marks)
Consider a system with 1KB (1024 bytes) page size. What are the page numbers and offsets for the following addresses?

Address	Page number	Offset
2375	2	327
19366	18	934
30000	29	304
256	0	256
16385	16	1

Q5 – Written question (5 marks)
Consider a system with a 32-bit logical address space and 4KB page size. The system supports up to 512MB of physical memory. How many entries are there in each of the following?
a) A conventional single-level page table.
total virtual memory size=2^32
page size = 4kb = 2^12
entries = 2^32/2^12 = 2^20 =1048576

b) An inverted page table.
total physical memory = 2^29=512MB
page size = 4kb = 2^12
entries = 2^29 / 2^12 =2^17 =131072

Q6 – Written question (5 marks)

Consider a system where a direct memory reference takes 200ns.
a) If we add a single-level page table stored in memory to this system, how much time
would it take to locate and reference a page in memory?
2*200 =400 ns
b) If we also add a TLB, and 75% of all page-table references are found in the TLB, what is the effective access time ? Assume that searching TLB takes 10ns.
0.75*(200+10)+0.25*(400+10) = 260ns

Q7 – Written question (5 marks)
Consider the following page reference string:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.
Assume there are 3 frames in the physical memory and all frames are initially empty. Illustrate how pages are placed into the frames according to the LRU and the optimal replacement algorithms. How many page faults would occur for each algorithm? Show your work.

LRU algorithm:

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

The grey columns stands for page faults, so the number of page fault is 15.

optimal algorithm:

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

The grey columns stands for page faults, so the number of page fault is 11.