109590004 呂育瑋 作業系統 HW3 hand-written part

7.6

–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, and 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?

* 1. (a) Increase Available (new resources added).
  2. (b) Decrease Available (resource permanently removed from system).
  3. (c) Increase Max for one process (the process needs or wants more resources than allowed).
  4. (d) Decrease Max for one process (the process decides that it does not need that many resources).
  5. (e) Increase the number of processes.
  6. (f) Decrease the number of processes.

7.12 Consider the following snapshot of a system : 一張含有 資料表 的圖片

自動產生的描述

–Use the banker’s algorithm , determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe.

(a) Available = ( 0, 3, 0, 1 )

(b) Available = ( 1, 0, 0, 2 )

7.15

– A single lane bridge connects the two Vermont villages of North Tunbridge and South Tunbridge. Farmers in the two villages use this bridge to deliver their produce to the neighbor town.

– The bridge can become deadlocked if a northbound and a southbound farmer get on the bridge at the same time. (Vermont farmers are stubborn and are unable to back up.)

– Using semaphores and/or mutex locks , design an algorithm in pseudocode that prevents deadlock.

– Initially, do not be concerned about starvation (the situation in which northbound farmers prevent southbound farmers from using the bridge, or vice versa).

8.5

–Compare the memory organization schemes of contiguous memory allocation, pure segmentation , and pure paging with respect to the following issues:

• (a) external fragmentation

• (b) internal fragmentation

• (c) ability to share code across processes

8.9

–Compare paging with segmentation with respect to how much memory the address translation structures require to convert virtual addresses to physical addresses.

8.13

–The BTV operating system has a 21 bit virtual address, yet on certain embedded devices, it has only a 16 bit physical address. It also has a 2KB page size. How many entries are there in each of the following?

– (a) A conventional, single level page table

– (b) An inverted page table

9.6

–Assume that we have a demand paged memory.

• The page table is held in registers.

• It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified.

• Memory access time is 100 nanoseconds.

–Assume that the page to be replaced is modified 70 percent of the time.

–What is the maximum acceptable page fault rate for an effective access time of no more than 200 nanoseconds?

9.8

Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1.

Assume demand paging with three frames, how many page faults would occur for the following replacement algorithms?

(a) LRU replacement

(b) FIFO replacement

(c) Optimal replacement

9.17

A page replacement algorithm should minimize the number of page faults. We can achieve this minimization by distributing heavily used pages evenly over all of memory, rather than having them compete for a small number of page frames. We can associate with each page frame a counter of the number of pages associated with that frame. Then, to replace a page, we can search for the page frame with the smallest counter.

– (a) Define a page replacement algorithm using this basic idea. Specifically address these problems:

• ( i ) What is the initial value of the

• (ii) When are counters increased?

• (iii) When are counters decreased?

• (iv) How is the page to be replaced selected?

– (b) How many page faults occur for your algorithm for the following reference string with four page frames?

• 1,2,3,4,5,3,4,1,6,7,8,7,8,9,7,8,9,5,4,5,4,2.

– (c) What is the minimum number of page faults for an optimal page replacement strategy for the reference string in part(b) with four page frames?

9.19

What is the cause of thrashing ? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?