Operating Systems

Jinghui Zhong （钟竞辉）

Office： B3-515

Email： jinghuizhong@scut.edu.cn

In this problem you are to compare reading a file using a single-threaded file server and a multithreaded server. It takes 12 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in the block cache. If a disk operation is needed, as is the case one-third of the time, an additional 75 msec is required, during which time the thread sleeps. How many requests/sec can the server handle if it is single threaded? If it is multithreaded?

作答

Chapter 2, p.17

In this problem you are to compare reading a file using a single-threaded file server and a multithreaded server. It takes 12 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in the block cache. If a disk operation is needed, as is the case one-third of the time, an additional 75 msec is required, during which time the thread sleeps. How many requests/sec can the server handle if it is single threaded? If it is multithreaded?

For a single threaded server: When there is no disk operation needed, it takes 12ms to handle a request. While for the cases when a disk operation is needed, 12 + 75 = 87ms is needed. Since there are 1/3 cases that involve a disk operation, an average time of 1/3 * 87 + 2/3 * 12 = 37ms is needed to handle one request. Therefore the server can handle 1000/37 ≈ 27 requests per second.

For a multiple threaded server: When a disk operation is needed for a request, the threads that don't sleep can still handle the requests that don't require a disk operation. So a non-sleep thread is then loaded to CPU and handle another request. So the time to handle a request is 12ms. Therefore the server can handle 1000/12 ≈ 83.3 requests per second.

(all the waiting for the disk is overlapped)

Measurements of a certain system have shown that the average process runs for a time T before blocking on I/O. A process switch requires a time S, which is effectively wasted (overhead). For round-robin scheduling with quantum Q, give a formula for the CPU efficiency for each of the following:

(a) $Q = \infty$
(b) $Q > T$
(c) $S < Q < T$
(d) $Q = S$
(e) Q nearly 0

作答

Chapter 2, p.43

43. Measurements of a certain system have shown that the average process runs for a time T before blocking on I/O. A process switch requires a time S, which is effectively wasted (overhead). For round-robin scheduling with quantum Q, give a formula for the CPU efficiency for each of the following:

(a) $Q = \infty$
(b) $Q > T$
(c) $S < Q < T$
(d) $Q = S$
(e) Q nearly 0

Answer:

For Q = infinity and Q > T, a certain process will never be preempted until it terminates. So the efficiency of CPU is T/(S+T).

For S < Q < T, after every Q time, S time is needed for context switch. So the efficiency is Q/(Q+S).

For Q = S, it is Q/(Q+S)=0.5.

For Q -> 0, Q/(Q+S) -> 0, so the efficiency approaches to 0.

45. Five batch jobs. A through E, arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead.

(a) Round robin.
(b) Priority scheduling.
(c) First-come, first-served (run in order 10, 6, 2, 4, 8).
(d) Shortest job first.

For (a), assume that the system is multiprogrammed, and that each job gets its fair share of the CPU. For (b) through (d), assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

Answer:

(a) C process terminates at 2*5 = 10 min. D terminates at 10+2*4=18 min. B terminates at 18+2*3=24 min. E terminates at 24+2*2=28 min. A terminates at 28+2=30 min. The average turnaround time for each process is (10+18+24+28+30)/5=22min.

(b) According to the priority, B is the first one to be executed and it terminates at 6min. And then E terminates at 6+8=14 min. A terminates at 14+10=24 min. C terminates at 24+2=26 min. Finally D terminates at 26+4=30 min. The average turnaround time for each process is (6+14+24+26+30)/5=20 min.

(c) If jobs are executed A-E. A terminates at 10 min. B terminates at 10+6=16 min. C terminates at 16+2=18 min. D terminates at 18+4=22 min. E terminates at 22+8=30 min. The average turnaround time is (10+16+18+22+30)/5=19.2 min.

(d) The jobs are executed according to the running time. So the average time is (2*5 + 4*4 + 6*3 + 8*2 + 10*1)/5 = 14 min.

4. Consider a swapping system in which memory consists of the following hole sizes in memory order: 10 MB, 4 MB, 20 MB, 18 MB, 7 MB, 9 MB, 12 MB, and 15 MB. Which hole is taken for successive segment requests of

(a) 12 MB
(b) 10 MB
(c) 9 MB

for first fit? Now repeat the question for best fit, worst fit, and next fit.

Answer： Suppose a, b, c requests are subsequent：

First fit： (a) 20MB, (b) 10MB, (c) 18MB

Best fit： (a) 12MB, (b) 10MB, (c) 9MB

Worst fit： (a) 20MB, (b) 18MB, (c) 15MB

Next fit： (a) 20MB, (b) 18MB, (c) 9MB

11. Consider the following C program:

```
int     X[N];
int step = M;      /* M is some predefined constant */
for (int i = 0; i < N; i += step) X[i] = X[i] + 1;
```

(a) If this program is run on a machine with a 4-KB page size and 64-entry TLB, what values of M and N will cause a TLB miss for every execution of the inner loop?

(b) Would your answer in part (a) be different if the loop were repeated many times? Explain.

Chapter 3

11. Consider the following C program:

```
int     X[N];
int step = M;       /* M is some predefined constant */
for (int i = 0; i < N; i += step) X[i] = X[i] + 1;
```

(a) If this program is run on a machine with a 4-KB page size and 64-entry TLB, what values of M and N will cause a TLB miss for every execution of the inner loop?

(b) Would your answer in part (a) be different if the loop were repeated many times? Explain.

Answer:

Suppose sizeof(int) = 4B

(a) M has to be at least 1K to ensure a TLB miss for every access to an element of X. Since N affects only how many times X is accessed, any value of N is OK.

(b) M should still be at least 1K to ensure a TLB miss for every access to an element of X. But now N should be greater than 64*M to thrash the TLB.

36. A computer has four page frames. The time of loading, time of last access, and the *R* and *M* bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

(a) Which page will NRU replace?
(b) Which page will FIFO replace?
(c) Which page will LRU replace?
(d) Which page will second chance replace?

作答

Chapter 3

36. A computer has four page frames. The time of loading, time of last access, and the *R* and *M* bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

(a) Which page will NRU replace?
(b) Which page will FIFO replace?
(c) Which page will LRU replace?
(d) Which page will second chance replace?

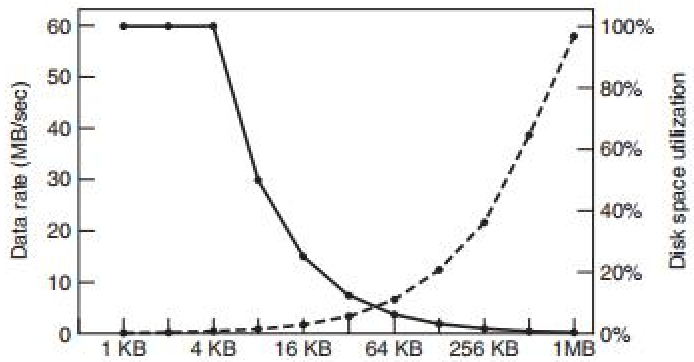
Answer:

(a) Page 2, because both R and M bit are 0
(b) Page 3, because it loaded at 110 time, which is the earliest.
(c) Page 1, because its last reference at 265, which is the earliest.
(d) Page 2

主观题 10分

设置

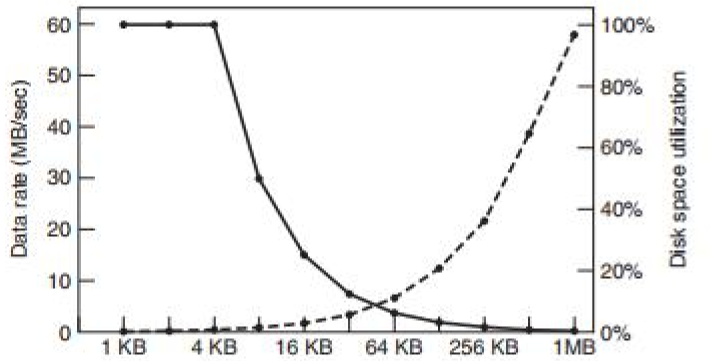
36. Consider the idea behind Fig. 4-21, but now for a disk with a mean seek time of 6 msec, a rotational rate of 15,000 rpm, and 1,048,576 bytes per track. What are the data rates for block sizes of 1 KB, 2 KB, and 4 KB, respectively?



作答

Chapter 4

36. Consider the idea behind Fig. 4-21, but now for a disk with a mean seek time of 6 msec, a rotational rate of 15,000 rpm, and 1,048,576 bytes per track. What are the data rates for block sizes of 1 KB, 2 KB, and 4 KB, respectively?



Answer:

At 15,000 rpm, the disk takes 4 msec to go around once. The average access time (in msec) to read k bytes is then $6 + 2 + (k/1,048,576) \times 4$.

For blocks of 1 KB, 2 KB, and 4 KB, the access times are about 8.0039 msec, 8.0078 msec, and 8.0156 msec, respectively.

These give rates of about 124.939 KB/sec, 255.751 KB/sec, and 511.004 KB/sec, respectively

38. Given a disk-block size of 4 KB and block-pointer address value of 4 bytes, what is the largest file size (in bytes) that can be accessed using 10 direct addresses and one indirect block?

作答

Chapter 4

38. Given a disk-block size of 4 KB and block-pointer address value of 4 bytes, what is the largest file size (in bytes) that can be accessed using 10 direct addresses and one indirect block?

The indirect block can hold 1024 addresses. Added to the 10 direct addresses, there are 1034 addresses in all. Since each one points to a 4-KB disk block, the largest file is 4136KB.

主观题 10分

设置

18. A disk rotates at 7200 RPM. It has 500 sectors of 512 bytes around the outer cylinder. How long does it take to read a sector?

作答

Chapter 5

18. A disk rotates at 7200 RPM. It has 500 sectors of 512 bytes around the outer cylinder. How long does it take to read a sector?

It rotates 7200/60 = 120 times per second. So it require 1/120*1000=8.33ms for one rotation. So reading a sector needs 8.33/500 = 0.0167ms.

主观题 10分

设置

38. A computer uses a programmable clock in square-wave mode. If a 500 MHz crystal is used, what should be the value of the holding register to achieve a clock resolution of

(a) a millisecond (a clock tick once every millisecond)?
(b) 100 microseconds?

作答

Chapter 5

38. A computer uses a programmable clock in square-wave mode. If a 500 MHz crystal is used, what should be the value of the holding register to achieve a clock resolution of

(a) a millisecond (a clock tick once every millisecond)?
(b) 100 microseconds?

Answer:
(1) 500MHz/1ms = 500,000. So the register needs holding 500,000.
500MHz/100us = 50,000. The register needs holding 50,000.

Session 1

【Objective and Requirement】

Objective: Be familiar with the creation of process and thread.

Requirement:

Task 1: Create a console application, “child”, which keeps printing out “The child is talking at [system time]” (in a loop, one per 1s).

Task 2: Create another console application, “parent”. It create a child **process** to execute “child”. At the same time, the “parent” process keeps printing out “The parent is talking at [system time]”. (one per 1s). Execute “parent” and explain the output you see.

Task 3: Create a child thread in the “mainThread” program. Both the main thread and the child thread keep printing out “[ThreadID] + [System time]”.

Task 4: Create a console application, which contains a shared integer shared_var. The initial value of shared_var is 0. The application will create a child **thread** after it starts. The main thread keeps increasing the value of shared_var by 1, while the child thread keeps decreasing the value of shared_var by 1. Explain the observed results.

【Environment】

Operating System: Linux;