

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN: NHẬP MÔN HỆ ĐIỀU HÀNH

**GIẢI ĐÁP CÁC PROBLEM
CHƯƠNG 5 (PHẦN SỐ CHẴN)
MODERN OPERATING SYSTEMS**

Người hướng dẫn: G.V Trần Trung Tín

Người thực hiện: Trần Quốc Lĩnh - 51703124

Lớp: 17050301

Khoá: 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2018

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN: NHẬP MÔN HỆ ĐIỀU HÀNH

GIẢI ĐÁP CÁC PROBLEM
CHƯƠNG 5 (PHẦN SỐ CHẴN)
MODERN OPERATING SYSTEMS

Người hướng dẫn: G.V Trần Trung Tín

Người thực hiện: Trần Quốc Linh - 51703124

Lớp: 17050301

Khoá: 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2018

LỜI CẢM ƠN

Cảm ơn thầy Trần Trung Tín đã tận tâm dạy môn nhập môn hệ điều hành, giúp em hiểu thế nào là hệ điều hành và cung cấp những kiến thức cần thiết để em làm bài tập lớn này!

Trong quá trình thực hiện bài tập lớn, em đã gặp không ít khó khăn. Cảm ơn thầy đã luôn dành thời gian quý báu của mình để tiếp sinh viên, để giúp những bạn như em giải quyết các khó khăn đó. Tuy nhiên vẫn không tránh khỏi những sai sót, em rất mong nhận được sự đánh giá và chỉ bảo từ thầy!

BÀI TẬP LỚN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Em xin cam đoan đây là sản phẩm bài tập lớn của riêng em. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu, hình ảnh được chính em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài tiểu luận còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung bài tập lớn của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 10 tháng 10 năm 2018

Tác giả

(Đã ký)

Trần Quốc Lĩnh

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày 10 tháng 10 năm 2018
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày 10 tháng 10 năm 2018
(kí và ghi họ tên)

TÓM TẮT

Với sự phát triển của thời đại kỹ thuật số, thì sở hữu một chiếc máy tính không còn là chuyện khó khăn nữa. Nhưng ít ai biết rằng để vận hành được một chiếc máy tính như thế đòi hỏi người ta phải tạo ra một chương trình, một hệ thống to lớn và phức tạp đến nhường nào. Bài tập lớn này sẽ trình bày một phần khá quan trọng trong hệ thống đồ sộ ấy, chính là **Input/Output**, thông qua việc giải quyết các problem (phần số chẵn) của chương 5, sách Modern operating systems.

Ngoài ra, em còn lên thư viện để tìm thêm thông tin để làm bài tập lớn vào các ngày 18/09, 02/10, 27/10, 30/10 và 05/11.

MỤC LỤC

LỜI CẢM ƠN	3
CAM KẾT	4
ĐÁNH GIÁ CỦA GIÁO VIÊN	5
TÓM TẮT	6
MỤC LỤC	7
DANH MỤC CHÚ THÍCH CÁC THUẬT NGỮ VÀ HÌNH ẢNH	8
GIẢI QUYẾT CÁC PROBLEM	
CHƯƠNG 5 - INPUT/OUTPUT (PHẦN SỐ CHẴN)	10
TÀI LIỆU THAM KHẢO	19

DANH MỤC CHÚ THÍCH CÁC THUẬT NGỮ VÀ HÌNH ẢNH

Thuật ngữ

BitBlt : Là một data operation thường được sử dụng trong đồ họa máy tính, trong đó một số bitmap được kết hợp thành một hàm boolean.

Bitmap : Là một ánh xạ từ một số miền thành các bit. Nó cũng được gọi là một mảng các bit hoặc bitmap index.

Cylinder : Tập hợp các track được sắp theo cùng một thứ tự.

DMA : Là một cơ chế cho phép chuyển dữ liệu trực tiếp giữa các thiết bị nhập/xuất và RAM mà không cần thông qua CPU.

ECC : (Error-Correcting Code) được sử dụng để kiểm soát lỗi dữ liệu trên các kênh không đáng tin cậy hoặc dễ bị nhiễu.

Interrupt-driven : Mỗi ngắt có trình xử lý ngắt của riêng nó. Số lượng ngắt của phần cứng bị giới hạn với số yêu cầu ngắt tới bộ xử lý, nhưng có thể có hàng trăm phần mềm ngắt khác nhau. Ngắt là một kỹ thuật dùng trong đa nhiệm, đặc biệt là trong thời gian thực. Hệ thống như vậy được gọi là interrupt-driven.

I/O : (Input/output) nhập/xuất.

Level : Mức độ.

NVRAM : (Non-Volatile Random Access Memory) không bị mất dữ liệu khi tắt nguồn. NVRAM lưu trữ tất cả các thông tin mà hệ thống phải có trước lúc khởi động.

Overhead : Là sự kết hợp giữa sự thừa hoặc thời gian tính toán gián tiếp, bộ nhớ, băng thông hoặc các tài nguyên khác được yêu cầu để thực hiện một tác vụ cụ thể.

Program counter : Là thanh ghi bộ xử lý cho biết máy tính đang ở đâu trong chuỗi chương trình của nó.

PSW : (Program status word) là một hệ thống kế thừa kiến trúc IBM system/360 và thanh ghi điều khiển cho phép thực hiện chức năng của thanh ghi trạng thái và bộ đếm của chương trình trong các kiến trúc khác và hơn thế nữa.

RAID : (Redundant Arrays of Inexpensive Disks hoặc Redundant Arrays of Independent Disks) là hình thức ghép nhiều ổ đĩa cứng vật lý thành một hệ thống ổ đĩa cứng, có chức gia tăng tốc độ đọc/ghi dữ liệu, hoặc nhằm tăng thêm sự an toàn của dữ liệu chứa trên hệ thống đĩa, hoặc kết hợp cả hai yếu tố trên.

Parity : Chẵn lẻ.

RGB : Sử dụng mô hình ánh sáng bổ sung trong đó có ánh sáng đỏ (Red), xanh lục (Green) và xanh lam (Blue) được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu khác.

RPM : (Revolutions per minute) là số vòng quay/phút.

Sector : Là đơn vị format nhỏ nhất trên đĩa.

Seek time : Thời gian di chuyển từ đầu đọc tới vị trí của dữ liệu trên đĩa.

SIGINT : Signals intelligence.

Stack : Ngăn xếp.

Track : Các vòng tròn trên đĩa.

Word : Một đơn vị dữ liệu kiểu word.

Hình ảnh

Hình 1 : Bảng liệt kê tốc độ của một số thiết bị, mạng và bus điển hình.

Hình 2 : Minh họa gián đoạn service procedure.

Hình 3 : Phân tích ảnh hưởng của sự cố đối với việc ghi ổn định

GIẢI QUYẾT CÁC PROBLEM

CHƯƠNG 5 - INPUT/OUTPUT (PHẦN SỐ CHẴN)

2. Với các tốc độ được liệt kê trong hình 5-1, có khả thi hay không khi quét (scan) tài liệu từ máy quét (scanner) và truyền chúng qua một mạng 802.11g với tốc độ tối đa?

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Scanner at 300 dpi	1 MB/sec
Digital camcorder	3.5 MB/sec
4x Blu-ray disc	18 MB/sec
802.11n Wireless	37.5 MB/sec
USB 2.0	60 MB/sec
FireWire 800	100 MB/sec
Gigabit Ethernet	125 MB/sec
SATA 3 disk drive	600 MB/sec
USB 3.0	625 MB/sec
SCSI Ultra 5 bus	640 MB/sec
Single-lane PCIe 3.0 bus	985 MB/sec
Thunderbolt 2 bus	2.5 GB/sec
SONET OC-768 network	5 GB/sec

Hình 1 : Bảng liệt kê tốc độ của một số thiết bị, mạng và bus điển hình.
 Nguồn: Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014)

Trả lời:

Khả thi! Vì máy quét (scanner) có tốc độ tối đa là 400KB/giây. Mà mạng không dây chạy ở tốc độ 6.75MB/giây nên sẽ không thành vấn đề khi truyền dữ liệu từ máy quét (scanner) qua mạng với tốc độ tối đa.

4. Giải thích sự cân bằng giữa precise interrupt và imprecise interrupt trên máy superscalar.

Trả lời:

Imprecise interrupt làm cho code phức tạp, mà ưu thế của precise interrupt là sự đơn giản của code trong hệ điều hành vì trạng thái của máy cũng được xác định. Tuy nhiên, precise interrupt làm tăng độ phức tạp của thiết kế chip và vùng chip, điều này có thể khiến CPU chậm hơn.

6. Giả sử rằng một hệ thống sử dụng DMA để truyền dữ liệu từ bộ điều khiển ổ đĩa sang bộ nhớ chính. Và giả sử trung bình mất t_1 giây để kết nối được với các bus và t_2 giây để chuyển một word qua bus ($t_1 \gg t_2$). Sau khi CPU đã được lập trình bộ điều khiển DMA, mất bao lâu để chuyển 1000 word từ bộ điều khiển ổ đĩa sang bộ nhớ chính, nếu

(a) Dùng chế độ word-at-a-time.

(b) Dùng chế độ burst. Giả sử rằng việc ra lệnh cho bộ điều khiển ổ đĩa yêu cầu kết nối tới bus để gửi đi một word và thừa nhận một lần truyền dữ liệu cũng yêu cầu bus gửi đi một word.

Trả lời:

(a) Chế độ word-at-time:

$$1000 \times [(t_1 + t_2) + (t_1 + t_2) + (t_1 + t_2)]$$

Nguồn: Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014).

Trong đó, phần đầu để kết nối với bus và gửi lệnh đến bộ điều khiển ổ đĩa, phần thứ hai để chuyển word, và phần thứ ba để được thừa nhận. Tất cả mất $3000 \times (t_1 + t_2)$ nano giây.

(b) Chế độ burst:

$$t_1 + t_2 + t_1 + \dots t_2 + t_1 + t_2 \text{ (1000 lần)}$$

Nguồn: Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014).

Trong đó, phần đầu dùng để kết nối tới bus và gửi lệnh tới bộ điều khiển đĩa, phần hai dùng để bộ điều khiển đĩa kết nối tới bus, phần ba để chuyển burst, và phần bốn để kết nối với bus và thực hiện thừa nhận. Tất cả mất $3t_1 + 1002t_2$.

8. Giả sử rằng máy tính có thể đọc hoặc ghi một memory word trong 5 nano giây. Cũng giả sử rằng khi có một sự gián đoạn xảy ra, tất cả 32 thanh ghi CPU, cộng thêm program counter và PSW được đẩy lên stack. Cho biết số lượng tối đa gián đoạn/giây mà máy tính này có thể xử lí?

Trả lời:

Một gián đoạn yêu cầu đưa 34 word vào stack. Gián đoạn khác yêu cầu nạp 34 word từ stack. Nó overhead 340 nano giây. Do đó, máy tính có thể xử lí tối đa không hơn 2,94 triệu gián đoạn/giây.

10. Trong hình 5-9(b), sự gián đoạn không được thừa nhận cho đến khi ký tự tiếp theo được đưa ra máy in. Nó có thể được thừa nhận ngay khi bắt đầu sự gián đoạn service procedure hay không? Nếu có, hãy đưa ra một lí do để làm điều đó ở bước cuối. Nếu không thì tại sao?

```
if (count == 0) {
    unblock_user();
} else {
    *printer_data_register = p[i];
    count = count - 1;
    i = i + 1;
}
acknowledge_interrupt();
return_from_interrupt();
```

Hình 2 : Minh họa gián đoạn service procedure

Nguồn: Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014)

Trả lời:

Có! Là vì đoạn code cuối của gián đoạn service procedure rất ngắn. Bởi lần đầu xuất ra một ký tự khác và sau đó thừa nhận gián đoạn, nếu một gián đoạn khác xảy ra ngay lập tức, máy in sẽ hoạt động suốt gián đoạn, làm cho nó in nhanh hơn một chút. Một bất lợi của phương pháp này là thời gian chết lâu hơn một chút khi các gián đoạn khác có thể bị vô hiệu hóa.

12. Một trang văn bản thường gồm 50 dòng và 80 ký tự mỗi dòng. Tưởng tượng rằng một máy in có thể in 6 trang/phút và thời gian để ghi một ký tự vào thanh ghi output của máy in là quá ngắn mà nó có thể bị bỏ qua. Hợp lí hay không khi chạy máy in này bằng cách sử dụng interrupt-driven I/O nếu mỗi ký tự được in yêu cầu một gián đoạn mất 50 μ giây cho tất cả cho dịch vụ đó?

Trả lời:

Máy in in $50 \times 80 \times 6 = 24\,000$ ký tự/phút = 400 ký tự/giây. Mỗi ký tự sử dụng 50 μ giây của CPU cho mỗi gián đoạn, vì vậy cùng với mỗi gián đoạn overhead 20 mili giây. Sử dụng interrupt-driven I/O, thời gian còn lại là 980 mili giây sẵn có cho nhu cầu khác. Nói cách khác, gián đoạn overhead tốn 2% của CPU, điều này hầu như không ảnh hưởng đến các chương trình đang chạy.

14. Mỗi tác vụ dưới đây được thực thi ở đâu trong bốn tầng phần mềm I/O

- (a) Tính toán track, sector và head cho việc đọc đĩa.
- (b) Ghi lệnh vào thanh ghi của thiết bị.
- (c) Kiểm tra xem liệu người dùng có được phép sử dụng thiết bị.
- (d) Chuyển đổi số nhị phân sang mã ASCII để in.

Trả lời:

- (a) Trình điều khiển thiết bị.
- (b) Trình điều khiển thiết bị.
- (c) Phần mềm độc lập với phần cứng.
- (d) Phần mềm người sử dụng.

16. Tại sao tập tin output cho máy in thường được đặt trên đĩa trước khi được in ra?

Trả lời:

Nếu máy in được gán ngay khi có output, một tiến trình có thể kết nối với máy in ngay lập tức bằng cách in một vài ký tự.

18. Một đĩa quay tại 700 RPM. Nó có 500 sector của 512 byte xung quanh outer cylinder. Mất bao lâu để đọc một sector?

Trả lời:

Tại 700 RPM, tức là 120 vòng/giây, vì vậy mất khoảng 8.33 mili giây. Chia cho 500, ta có được khoảng 16,67 μ giây để đọc một sector.

20. RAID level 3 có thể sửa lỗi bit đơn mà chỉ sử dụng một ổ đĩa parity. Còn RAID level 2 thì sao? Dù vậy, nó cũng chỉ có thể sửa chữa một lỗi duy nhất và tốn nhiều ổ đĩa để làm việc đó.

Trả lời:

RAID level 2 không chỉ có thể phục hồi từ các sự cố drive, mà còn từ các lỗi chưa được phát hiện. Nếu một drive hỏng một bit đơn duy nhất, RAID level 2 sẽ sửa lỗi này, mà RAID level 3 sẽ không.

22. So sánh RAID từ level 0 đến 5 về phương diện hiệu suất đọc, ghi, space overhead và độ tin cậy.

Trả lời:

Hiệu suất đọc: Các RAID level 0, 2, 3, 4, và 5 cho phép đọc song song để phục vụ yêu cầu đọc. Tuy nhiên, RAID level 1 cho phép hơn 2 yêu cầu đọc thực hiện đồng thời.

Hiệu suất ghi: Tất cả level RAID đều cung cấp hiệu suất ghi như nhau.

Space overhead: Không có space overhead trong level 0 và 1. Với 32-bit dữ liệu word và sáu parity drive, space overhead là khoảng 18.75% trong level 2, và 3.13% ở level 3, 4 và 5.

Độ tin cậy: Không hỗ trợ độ tin cậy ở level 0. Tất cả các level còn lại có thể an toàn sau một sự cố ổ đĩa. Thêm nữa, trong level 3, 4, và 5, và một lỗi bit đơn ngẫu nhiên trong một word có thể được phát hiện, và còn có thể được sửa chữa trong level 2.

24. Tại sao các thiết bị lưu trữ quang vốn có khả năng chứa mật độ dữ liệu cao hơn các thiết bị lưu trữ từ tính?

Trả lời:

Khi một vùng từ trường được phát ra giữa hai cực. Không chỉ khó khăn ở việc làm cho nguồn của vùng từ trường nhỏ, mà còn phải phân tán nhanh chóng, dẫn đến các vấn đề cơ học là khá khó để giữ cho bề mặt của từ trường medium gần với nguồn từ trường hoặc cảm biến. Một laser bán dẫn phát ra được ánh sáng ở một vùng rất nhỏ, và ánh sáng có thể được xử lý quang học để chiếu sáng ở một điểm rất nhỏ từ một khoảng cách tương đối lớn từ nguồn.

26. Nếu bộ điều khiển ổ đĩa ghi các byte nó nhận được từ ổ đĩa vào bộ nhớ nhanh như cách nó nhận chúng, mà không có bộ đệm nội bộ nào, là xen kẽ hữu ích? Giải thích câu trả lời.

Trả lời:

Có thể. Nếu hầu hết các tập tin được lưu trữ trong các sector luận lý liên tục, nó có thể là xen kẽ hữu ích đối với các sector để cung cấp cho các chương trình thời gian xử lý dữ liệu vừa được nhận, vậy nên khi yêu cầu tiếp theo được đưa ra, ổ đĩa sẽ ở đúng nơi. Cho dù đây là vấn đề phụ thuộc chủ yếu vào loại chương trình và cách đồng bộ của chúng.

28. Xét một đĩa từ gồm 16 head và 400 cylinder. Ổ đĩa này có bốn khu vực với 100 cylinder mỗi khu vực, bên trong chứa 160, 200, 240 và 280 sector tương ứng. Giả sử rằng mỗi sector chứa 512 byte, seek time trung bình giữa các cylinder liên kế là 1 mili giây, và ổ đĩa quay ở tốc độ 7200 RPM. Tính

- (a) Dung lượng ổ đĩa
- (b) Optimal track skew
- (c) Tốc độ truyền dữ liệu tối đa.

Trả lời:

- (a) Ta có: Sức chứa của một vùng = $track \times cylinder \times sector / cylinder \times byte/sec$

Nguồn: Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014).

Sức chứa của vùng 1: $16 \times 100 \times 160 \times 512 = 131072000$ byte

Sức chứa của vùng 2: $16 \times 100 \times 200 \times 512 = 163840000$ byte

Sức chứa của vùng 3: $16 \times 100 \times 240 \times 512 = 196608000$ byte

Sức chứa của vùng 4: $16 \times 100 \times 280 \times 512 = 229376000$ byte

Tổng = $131072000 + 163840000 + 196608000 + 229376000 = 720896000$ byte

- (b) Tốc độ quay 7200 tức là 120 vòng/giây. Trong seek time track-to-track 1 mili giây, 0.12 của sector được cover.

Trong khu vực 1, head của ổ đĩa sẽ băng qua 0.12×160 sector trong 1 mili giây, vậy nên, optimal track skew đối với khu vực 1 là 19.2 sector.

Trong khu vực 2, head của ổ đĩa sẽ băng qua 0.12×200 sector trong 1 mili giây, vì vậy, optimal track skew đối với khu vực 2 là 24 sector.

Trong khu vực 3, head của ổ đĩa sẽ băng qua 0.12×240 sector trong 1 mili giây, vậy, optimal track skew đối với khu vực 3 là 28.8 sector.

Trong khu vực 4, head của ổ đĩa sẽ băng qua 0.12×280 sector trong 1 mili giây, nên, optimal track skew của khu vực 4 là 33.6 sector.

- (c) Tốc độ truyền dữ liệu đạt tối đa là khi cylinder ở khu vực ngoài cùng (khu vực 4) đang được đọc/ghi. Trong khu vực đó, trong một giây, 280 sector được đọc 120 lần. Do đó, tốc độ truyền dữ liệu tối đa là $280 \times 120 \times 512 = 17\ 203\ 200$ byte/giây.

30. Một nhà sản xuất máy tính quyết định thiết kế lại bảng phân khu của đĩa cứng Pentium để cung cấp nhiều hơn bốn phân khu. Một số hệ quả của sự thay đổi này là?

Trả lời:

Một thực tế rõ ràng là không tồn tại hệ điều hành nào hoạt động được bởi chúng đều nhìn vào đó để xem các phân vùng ổ đĩa ở đâu. Thay đổi định dạng của bảng phân vùng sẽ khiến tất cả các hệ điều hành ngừng hoạt động. Chỉ có một cách duy nhất để thay đổi bảng phân vùng là đồng thời thay đổi tất cả các hệ điều hành để sử dụng định dạng mới.

32. Một sửa đổi nhỏ của thuật toán thang máy đối với yêu cầu định thời ổ đĩa là luôn scan trong cùng một hướng. Thuật toán sửa đổi này có gì tốt hơn so với thuật toán thang máy?

Trả lời:

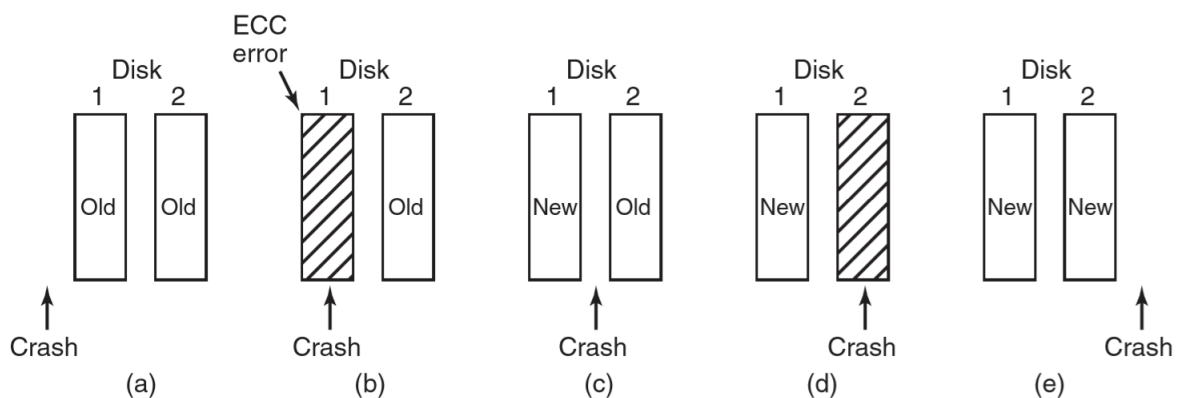
Trong trường hợp xấu nhất, một yêu cầu đọc/ghi không được đáp ứng cho gần hai lần scan toàn bộ ổ đĩa trong thuật toán thang máy, trong khi tối đa là một lần scan toàn bộ ổ đĩa cho thuật toán sửa đổi.

34. Điều gì xảy ra nếu một sự cố xảy ra trước khi hệ điều hành kịp ghi lại một số block không hợp lệ trên NVRAM? Điều này có làm hỏng sự trừu tượng của ổn định lưu trữ? Giải thích?

Trả lời:

Không! Thực tế là một số thông tin chưa được ghi lại trên NVRAM làm cho chương trình phục hồi sẽ không biết block nào đã được ghi lại. Nó sẽ đọc cả hai bản copy, tìm điểm giống nhau giữa chúng, xem xét có sự thay đổi không, sau đó chọn 1 hành động chính xác. Ảnh hưởng của sự cố trước khi NVRAM được cập nhật chỉ làm cho chương trình phục hồi làm việc nhiều hơn cần thiết.

36. Giả định rằng CPU làm hỏng một khu vực nào đó dẫn đến một ECC hoạt động không chính xác. Chuyện gì có thể phát sinh trong năm cơ chế phục hồi sự cố đã được trình bày ở hình 5-27 nếu giả thuyết này không đúng?



Hình 3 : Phân tích ảnh hưởng của sự cố đối với việc ghi ổn định

Nguồn: Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014)

Trả lời:

Nếu ECC thật sự hoạt động không chính xác, sẽ không thể phát hiện ở đĩa nào chứa lock hợp lệ (cũ hay mới), và cũng không thể phục hồi được dữ liệu.

38. Một máy tính dùng một clock lập trình trong chế độ sóng vuông. Nếu dùng crystal 500 MHz, giá trị của thanh ghi là bao nhiêu để clock

chuyển sang trạng thái khác mỗi

(a) Một mili giây (clock điểm từng mili giây)?

(b) 100 μ giây?

Trả lời:

Ta có:

(a) Sử dụng crystal 500 MHz, bộ đếm có thể giảm mỗi 2 nano giây. Nên, để đếm được mỗi mili giây, giá trị trên thanh ghi phải là $1\,000\,000/2 = 500\,000$.

(b) Để một clock điểm mỗi 100 μ giây, giá trị trên thanh ghi phải là 50 000.

40. Nhiều phiên bản của UNIX sử dụng số nguyên 32-bit unsigned để theo dõi số giây kể từ thời gian gốc. Lần tiếp theo hệ thống wrap around là khi nào (năm và tháng)? Điều này có khả năng xảy ra hay không?

Trả lời:

Số giây trong một năm: $365,25 \times 24 \times 3\,600 = 31\,557\,600$. Bộ đếm đã wrap around sau 232 giây từ 1/1/1970. Vậy nên việc wrapping sẽ xảy ra tiếp theo khoảng 136,1 năm nữa tức là vào tháng 2 năm 2106. Và khi đó mỗi máy tính sẽ có ít nhất 64bit, vậy nên điều này không có khả năng xảy ra.

42. Sau khi nhận được kí tự DEL (SIGNINT), trình điều khiển hiển thị loại bỏ tất cả output trong hàng đợi hiện tại cho hiển thị đó? Tại sao?

Trả lời:

Giả sử người dùng vô tình yêu cầu trình soạn thảo in hàng ngàn dòng. Sau đó, người dùng ấn DEL để dừng điều đó lại. Nếu trình điều khiển hiển thị không loại bỏ tất cả output trong hàng đợi ngay lập tức sẽ khiến cho người dùng khó chịu khi không thấy có gì xảy ra và họ có thể sẽ ấn liên tục vào DEL.

44. Các nhà thiết kế hệ thống máy tính mong đợi rằng những con chuột có thể di chuyển với tốc độ tối đa 20cm/giây. Nếu kích thước là 0.1 mm và mỗi con chuột truyền tin 3 byte, thì tốc độ truyền dữ liệu tối đa của con chuột là bao nhiêu? Giả định là mỗi con chuột được report một cách riêng biệt.

Trả lời:

Tốc độ tối đa mà chuột máy tính có thể di chuyển là 200mm/giây, tức là 2 000 mickey/giây. Nếu mỗi report là 3 byte, tốc độ đầu ra sẽ là 6 000 byte/giây.

46. Một phương pháp để xếp chỗ một ký tự trên màn hình bitmap là sử dụng BitBlt từ bản phông chữ. Giả sử một phông chữ cụ thể sử dụng

các ký tự có kích thước 16×24 pixel với mã màu RGB.

(a) Mỗi ký tự chiếm bao nhiêu không gian trên bản phong chữ?

(b) Nếu copy một byte mất 100 nano giây, bao gồm phần đầu, thì tốc độ output cho ra trên màn hình là bao nhiêu ký tự/giây?

Trả lời:

(a) Mỗi pixel tiêu tốn 3 byte trên mã màu RGB, nên không gian của bảng phải là $16 \times 24 \times 3 \text{ byte} = 1\,152 \text{ byte}$.

(b) Tại 100 nano giây mỗi byte, mỗi ký tự tiêu tốn $115,2 \mu$ giây. Nên, tốc độ đầu ra là 8 681 ký tự/giây.

TÀI LIỆU THAM KHẢO

Tiếng Việt

<https://vi.wikipedia.org/wiki/Bitmap>
https://vi.wikipedia.org/wiki/C%C6%A1_ch%E1%BA%BF_DMA
<https://vi.wikipedia.org/wiki/RAID>
<https://techgosu.com/nvram-la-gi-reset-nvram-khi-nao.html>

Tiếng Anh

<https://en.wikipedia.org/wiki/Input/output>
https://en.wikipedia.org/wiki/Input%E2%80%93output_model
Abraham Silberschatz, Peter B. Galvin, Greg Gagne, [2013], Operating System Concepts, 9th edition, Wiley.
Andrew S. Tanenbaum, Herbert Bos-Modern Operating Systems-Pearson (2014).
Operating Systems Design and Implementation (3rd Edition).