Phần mềm I/O (Input/Output) là các phần mềm được thiết kế để điều khiển và quản lý việc truyền dữ liệu giữa các thiết bị ngoại vi và hệ thống máy tính.

* Goals of the I/O Software
* Programmed I/O
* Interrupt-Driven I/O
* I/O using DMA

độc lập thiết bị

Viết chương trình có thể truy cập bất kỳ thiết bị I/O nào mà không cần phải chỉ định thiết bị

đặt tên thống nhất

Tên của tệp hoặc thiết bị chỉ nên là một chuỗi hoặc một số nguyên và không phụ thuộc vào thiết bị theo bất kỳ cách nào

Một tên đường dẫn được sử dụng để giải quyết

xử lý lỗi

Vấn đề không thể xử lý bằng phần cứng, nên được thông báo bằng phần mềm

Truyền đồng bộ (chặn) so với truyền không đồng bộ (điều khiển bằng ngắt)

Chương trình người dùng ghi nếu thao tác I/O bị chặn – sau khi hệ thống đọc lệnh gọi, chương trình sẽ tự động bị treo cho đến khi dữ liệu có sẵn trong bộ đệm

CPU bắt đầu truyền và tắt để làm việc khác cho đến khi có ngắt

đệm

Dữ liệu phải được đưa vào bộ đệm đầu ra trước để tách tốc độ bộ đệm được lấp đầy khỏi tốc độ bộ đệm trống, để tránh bộ đệm đang chạy

Chia sẻ so với thiết bị chuyên dụng: Xử lý hệ điều hành để tránh sự cố

I/O được lập trình: Hoạt động I/O có sự tham gia của CPU

|  |  |
| --- | --- |
| IO Method | Advantage(s) / Disadvantage(s) |
| Bỏ phiếu hoặc IO đang chờ bận:  CPU sử dụng tính năng bỏ phiếu để xem bit điều khiển của thiết bị và liên tục lặp lại để xem liệu thiết bị đã sẵn sàng chưa. | Đơn giản.  Buộc CPU toàn thời gian cho đến khi hoàn thành tất cả IO. |
| IO điều khiển ngắt: CPU nhận ngắt khi thiết bị sẵn sàng. | CPU có thể làm việc khác trong khi chờ thiết bị  Nếu bộ đệm không được sử dụng, ngắt xảy ra trên mọi ký tự  Hiệu suất có thể giảm. |
| IO Sử dụng DMA: Cho phép bộ điều khiển DMA nạp từng ký tự vào thiết bị một lần mà không làm phiền CPU. | Giải phóng CPU trong I/O để thực hiện công việc khác và giảm số lần ngắt và hầu hết thời gian DMA đều có giá trị.  Cần có phần cứng đặc biệt. |

Các bước:

Thực hiện cuộc gọi hệ thống để mở thiết bị

Nếu nó có sẵn, dữ liệu sẽ được sao chép vào kernel

Sau đó, CPU thăm dò thiết bị để xem nó đã sẵn sàng chấp nhận hay chưa

Nếu vậy, dữ liệu nhỏ được xuất ra

Các bước in chuỗi bằng IO điều khiển ngắt

Mã được thực thi tại thời điểm cuộc gọi hệ thống in được thực hiện.

Quy trình dịch vụ gián đoạn cho máy in

In chuỗi bằng DMA

Mã được thực thi khi cuộc gọi hệ thống in được thực hiện.

Thủ tục dịch vụ gián đoạn.

Thời gian cần thiết được xác định bởi 3 yếu tố:

Tìm kiếm thời gian (thời gian để di chuyển cánh tay đến hình trụ thích hợp)

Độ trễ quay (thời gian để khu vực thích hợp quay dưới đầu)

Thời gian truyền dữ liệu thực tế

Đối với hầu hết các đĩa, thời gian tìm kiếm chiếm ưu thế trong hai thời gian còn lại, do đó, việc giảm thời gian tìm kiếm trung bình có thể cải thiện đáng kể hiệu suất hệ thống.

Xử lý yêu cầu tuần tự

Về bản chất là công bằng, nhưng nó thường không cung cấp dịch vụ nhanh nhất

Giả sử rằng đầu đĩa ban đầu ở xi lanh 53

Hàng đợi đĩa có yêu cầu khối I/O trên trụ theo thứ tự

98, 183, 37, 122, 14, 124, 65, 67

Chức vụ: 53 98 183 37 122 14 124 65 67

Tổng chuyển động của đầu là 640 ((98-53)+(183-98)+(183-37)+(122-37)+(122-14)+ (124-14) + (124-65)+(67 -65))

Độ dài tìm kiếm trung bình: 640/8 = 80

Chọn yêu cầu I/O của đĩa yêu cầu ít chuyển động nhất của nhánh đĩa từ vị trí đầu hiện tại của nó

 Luôn chọn thời gian tìm kiếm nhỏ nhất

Giả sử rằng đầu đĩa ban đầu ở xi lanh 53

Một hàng đợi đĩa với các yêu cầu khối I/O trên các trụ theo thứ tự 98, 183, 37, 122, 14, 124, 65, 67

Chức vụ: 53 65 67 37 14 98 122 124 183

Tổng chuyển động của đầu là 236 ((65-53)+(67-65)+(67-37)+(37-14)+(98-14)+(122-98)+(122-124)+(183 -124))

Độ dài tìm kiếm trung bình: 236/8 = 29,5

Các vấn đề

Khi SSF đang tiến hành, yêu cầu mới sẽ được ưu tiên tổng thể hoặc lựa chọn tiếp theo sẽ xuất hiện. Cánh tay tiếp theo sẽ đi đến yêu cầu mới.

Với một đĩa được tải nặng, cánh tay sẽ có xu hướng ở giữa đĩa trong hầu hết thời gian, do đó, các yêu cầu ở một trong hai cực sẽ phải đợi cho đến khi một biến động thống kê (dao động bất thường) trong quá trình tải khiến không có yêu cầu gần giữa. → Yêu cầu xa tầm trung có thể nhận được dịch vụ kém. Các mục tiêu về thời gian phản hồi tối thiểu và sự công bằng đang mâu thuẫn ở đây.

Cánh tay chỉ di chuyển theo một hướng, đáp ứng tất cả các yêu cầu chưa xử lý cho đến khi đến đường đua cuối cùng theo hướng đó.

Sau khi tất cả các yêu cầu đĩa theo một hướng được phục vụ, hướng đó sẽ bị đảo ngược.

Yêu cầu phần mềm duy trì 1 bit: bit hướng hiện tại, LÊN hoặc XUỐNG

Khi một yêu cầu kết thúc, trình điều khiển đĩa hoặc thang máy sẽ kiểm tra bit.

Nếu nó là TĂNG, cánh tay hoặc cabin được chuyển đến yêu cầu đang chờ xử lý cao nhất tiếp theo

Nếu không có yêu cầu nào đang chờ xử lý ở các vị trí cao hơn, bit hướng sẽ bị đảo ngược

Khi bit được đặt thành XUỐNG, việc di chuyển sẽ đến vị trí yêu cầu thấp nhất tiếp theo, nếu có.

Giả sử rằng đầu đĩa ban đầu ở xi lanh 53

Hàng đợi đĩa có yêu cầu khối I/O trên trụ theo thứ tự

98, 183, 37, 122, 14, 124, 65, 67

Vị trí với bit là XUỐNG:

53 37 14 65 67 98 122 124 183

Tổng chuyển động của đầu là 208 ((53-37)+(37-14)+(65-14)+(67-65)+(98-67)+(122-98)+(124-122)+(183 -124))

Độ dài tìm kiếm trung bình: 208/8 = 26

Vị trí với bit là LÊN:

53 65 67 98 122 124 183 37 14

Tổng chuyển động của đầu là 299 ((65-53)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)+(183-37)+(37 -14))

Độ dài tìm kiếm trung bình: 299/8 = 37,375

Thuộc tính tuyệt vời của thuật toán thang máy: Với bất kỳ tập hợp yêu cầu nào, giới hạn trên của tổng chuyển động là cố định: nó chỉ gấp đôi số lượng hình trụ.

Một sửa đổi nhỏ của thuật toán thang máy có phương sai nhỏ hơn trong thời gian phản hồi là luôn quét theo cùng một hướng.

Khi cột được đánh số cao nhất có yêu cầu đang chờ xử lý đã được cung cấp dịch vụ, cánh tay sẽ chuyển đến cột số thấp nhất có yêu cầu đang chờ xử lý và sau đó tiếp tục di chuyển theo hướng đi lên.

Trên thực tế, xi lanh có số thấp nhất được coi là ngay phía trên xi lanh có số cao nhất

Giả sử rằng đầu đĩa ban đầu ở xi lanh 53

Hàng đợi đĩa có yêu cầu khối I/O trên trụ theo thứ tự

98, 183, 37, 122, 14, 124, 65, 67

Chức vụ: 53 65 67 98 122 124 183 14 37

Tổng chuyển động của đầu là 153(?) ((65-53)+(67-65)+(98-67)+(122-98)+(124-122)+(183-14)+(37-14) )

Độ dài tìm kiếm trung bình: 153/8 = 19,125

Một số bộ điều khiển đĩa cung cấp một cách để phần mềm kiểm tra số cung hiện tại dưới đầu

Nếu hai hoặc nhiều yêu cầu cho cùng một xi lanh đang chờ xử lý, người lái xe có thể đưa ra yêu cầu cho khu vực sẽ đi qua đầu tiếp theo

Bộ điều khiển có thể chọn bất kỳ đầu nào của nó gần như ngay lập tức (tức thời)

Nếu thời gian tìm kiếm nhanh hơn nhiều so với độ trễ quay

Yêu cầu đang chờ xử lý phải được sắp xếp theo số khu vực và ngay khi khu vực tiếp theo sắp đi qua đầu, cánh tay phải được kéo qua đúng đường để đọc hoặc ghi nó

Khi một số ổ đĩa có mặt trên cùng một bộ điều khiển, HĐH sẽ duy trì một bảng yêu cầu đang chờ xử lý cho từng ổ đĩa riêng biệt

Bất cứ khi nào bất kỳ ổ đĩa nào không hoạt động, một lệnh tìm kiếm sẽ được đưa ra để di chuyển cánh tay của nó đến xi lanh, nơi sẽ cần nó tiếp theo

Khi quá trình truyền hiện tại kết thúc, có thể thực hiện kiểm tra xem có bất kỳ ổ đĩa nào được định vị trên đúng xi lanh không

Nếu có một hoặc nhiều, lần truyền tiếp theo có thể được bắt đầu trên một ổ đĩa đã có trên xi lanh bên phải

Nếu không có nhánh nào ở đúng vị trí, người lái xe nên đưa ra lệnh tìm kiếm mới trên ổ đĩa vừa hoàn thành chuyển và đợi cho đến lần ngắt tiếp theo để xem nhánh nào đến đích trước

Tất cả các thuật toán lập lịch trình đĩa ngầm giả định rằng hình học đĩa thực giống như hình học ảo.

Nếu không, thì việc lên lịch các yêu cầu đĩa không có ý nghĩa gì vì HĐH thực sự không thể biết liệu xi lanh 40 hay xi lanh 200 gần xi lanh 39 hơn.

Mặt khác, nếu bộ điều khiển đĩa có thể chấp nhận nhiều yêu cầu chưa xử lý, thì nó có thể sử dụng các thuật toán lập lịch trình này trong nội bộ. Trong trường hợp đó, các thuật toán vẫn hợp lệ, nhưng giảm một cấp, bên trong bộ điều khiển.