



(12) **BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) 
1-0022033

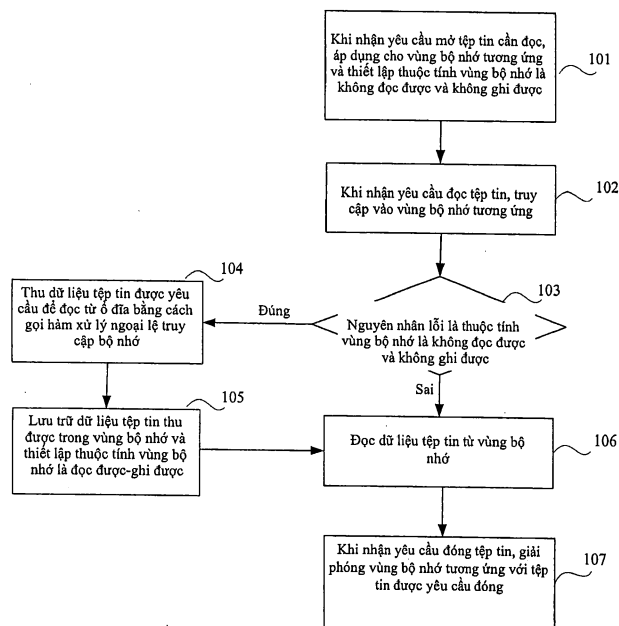
(51)⁷ **G06F 12/08**

(13) **B**

(21) 1-2014-02630 (22) 31.01.2013
(86) PCT/CN2013/071199 31.01.2013 (87) WO2013/117142 15.08.2013
(30) 201210025896.X 07.02.2012 CN
(45) 25.10.2019 379 (43) 25.12.2014 321
(73) Tencent Technology (Shenzhen) Company Limited (CN)
Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen,
Guangdong 518044, CHINA
(72) Zhuo CHEN (CN), Zipan BAI (CN)
(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) **PHƯƠNG PHÁP VÀ HỆ THỐNG XỬ LÝ TỆP TIN**

(57) Sáng chế đề cập đến phương pháp và hệ thống xử lý tệp tin. Phương pháp xử lý tệp tin có thể bao gồm bước nhận yêu cầu mở tệp tin, áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở và thiết lập thuộc tính của vùng bộ nhớ là không đọc được và không ghi được. Sau khi nhận yêu cầu đọc tệp tin, thì có thể truy cập vùng bộ nhớ tương ứng với tệp tin được yêu cầu đọc. Nếu việc truy cập có lỗi mà nguyên nhân lỗi là do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì dữ liệu tệp tin được yêu cầu đọc có thể thu được từ ổ đĩa bằng cách gọi hàm xử lý ngoại lệ truy cập bộ nhớ. Dữ liệu tệp tin thu được có thể được lưu trữ trong vùng bộ nhớ. Thuộc tính của vùng bộ nhớ có thể được thiết lập là đọc được và ghi được và dữ liệu tệp tin đọc được từ vùng bộ nhớ.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến lĩnh vực kỹ thuật xử lý tệp tin và cụ thể hơn đến phương pháp và hệ thống xử lý tệp tin.

Tình trạng kỹ thuật của sáng chế

API Windows (tức là giao diện lập trình ứng dụng) là hàm định trước của Windows, hàm này được sử dụng để kiểm soát hình thái và hoạt động của thành phần khác nhau trong hệ điều hành Windows. Thông qua các hàm API Windows, hệ điều hành Windows có thể được thông báo về các thao tác của người sử dụng. Tất cả các chương trình phần mềm ứng dụng trong hệ điều hành Windows có thể gọi là các hàm API Windows.

Hiện nay, hệ điều hành Windows định vị dữ liệu cần đọc bằng cách gọi hàm con trỏ thiết lập tệp tin (SetFilePointer) và sau đó đọc tệp tin bằng cách gọi hàm đọc tệp tin (ReadFile). Khi thường xuyên đọc một tệp tin, hệ điều hành Windows phải thường xuyên gọi hàm SetFilePointer để con trỏ tệp tin trở vào vùng của tệp tin cần đọc, và sau đó gọi hàm ReadFile để đọc dữ liệu trong vùng này. Mỗi lần đọc dữ liệu đòi hỏi thao tác tìm kiếm rãnh ghi trên ổ đĩa và, mỗi lần như vậy tạo ra các hoạt động IO của ổ đĩa (tức là đầu vào/đầu ra). Kết quả là, tốc độ đọc tệp tin bị giảm một cách bất lợi.

Do đó, cần có một phương pháp và hệ thống xử lý tệp tin để khắc phục các vấn đề này và vấn đề khác.

Bản chất kỹ thuật của sáng chế

Các phương án khác nhau của sáng chế đề xuất phương pháp và hệ thống xử lý tệp tin để giảm việc thường xuyên sinh ra các hoạt động IO của ổ đĩa khi đọc các tệp tin và để tiết kiệm các tài nguyên hệ thống.

Theo các phương án khác nhau, sáng chế đề xuất phương pháp xử lý tệp tin dựa vào hệ điều hành, bằng cách đăng ký hàm xử lý ngoại lệ truy cập bộ nhớ với hệ điều hành này, nhận yêu cầu mở tệp tin, áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở, và thiết lập thuộc tính của vùng bộ nhớ là không đọc được

và không ghi được. Sau khi nhận yêu cầu đọc tệp tin, thì có thể truy cập được vùng bộ nhớ. Nếu việc truy cập lỗi do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì có thể thu được dữ liệu tệp tin được yêu cầu đọc từ ổ đĩa bằng cách gọi hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký trong hệ điều hành. Dữ liệu tệp tin thu được có thể được lưu trữ trong vùng bộ nhớ. Thuộc tính của vùng bộ nhớ có thể được thiết lập là đọc được và ghi được và dữ liệu tệp tin đọc được từ vùng bộ nhớ.

Theo các phương án khác nhau, sáng chế đề xuất hệ thống xử lý tệp tin bao gồm môđun nhận, môđun xử lý thứ nhất, môđun xử lý thứ hai, và môđun thu. Môđun nhận có thể được tạo cấu hình để nhận yêu cầu mở tệp tin và để nhận yêu cầu đọc tệp tin. Môđun xử lý thứ nhất có thể được tạo cấu hình để áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở và để thiết lập thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, khi môđun nhận nhận yêu cầu mở tệp tin. Môđun xử lý thứ hai có thể được tạo cấu hình để truy cập vùng bộ nhớ, khi môđun nhận nhận yêu cầu đọc tệp tin và để thu dữ liệu tệp tin được yêu cầu đọc từ ổ đĩa bằng cách gọi hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký trong hệ điều hành. Môđun xử lý thứ hai còn được tạo cấu hình, nếu việc truy cập vào vùng bộ nhớ lỗi do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, để lưu trữ dữ liệu tệp tin thu được trong vùng bộ nhớ và để thiết lập thuộc tính của vùng bộ nhớ là đọc được và ghi được. Môđun thu có thể được tạo cấu hình để đọc dữ liệu tệp tin từ vùng bộ nhớ.

Theo một phương án cụ thể, phương pháp và hệ thống xử lý tệp tin của sáng chế có thể bao gồm bước đăng ký hàm xử lý ngoại lệ truy cập bộ nhớ trong một hệ điều hành, áp dụng cho vùng bộ nhớ tương ứng với tệp tin cần được mở, và đọc dữ liệu tệp tin từ vùng bộ nhớ. Nếu không truy cập được vào bộ nhớ do thuộc tính vùng bộ nhớ là không đọc được và không ghi được, thì có thể thu được dữ liệu tệp tin được yêu cầu đọc từ ổ đĩa bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ. Như được mô tả ở trên, khi dữ liệu tệp tin từ một vùng cụ thể được đọc, thì có thể tạo ra các hoạt động IO của ổ đĩa chỉ khi dữ liệu tệp tin được đọc lần thứ nhất. Sau đó, có thể lưu trữ dữ liệu tệp tin trong vùng bộ nhớ. Nếu sau đó cần đọc dữ liệu tệp tin một lần nữa từ cùng một vùng nhất định, thì có thể đọc trực tiếp dữ liệu tệp

tin từ vùng bộ nhớ. Không một IO nào của ổ đĩa có thể được tạo ra một lần nữa. Các tài nguyên của hệ thống có thể được tiết kiệm để cải thiện một cách hiệu quả việc đọc các tệp tin.

Các khía cạnh hoặc phương án khác của sáng chế sẽ được các chuyên gia trong lĩnh vực kỹ thuật này hiểu rõ thông qua phần mô tả chi tiết, yêu cầu bảo hộ và các hình vẽ kèm theo.

Mô tả vắn tắt các hình vẽ

Các hình vẽ dưới đây chỉ là các hình vẽ làm ví dụ nhằm minh họa sáng chế theo các phương án khác nhau và không nhằm giới hạn phạm vi của sáng chế.

Fig.1 là hình vẽ minh họa phương pháp xử lý tệp tin làm ví dụ theo các phương án khác nhau của sáng chế;

Fig.2 là hình vẽ minh họa phương pháp xử lý tệp tin làm ví dụ bằng cách gọi hàm xử lý ngoại lệ truy cập bộ nhớ theo các phương án khác nhau của sáng chế;

Fig.3 là hình vẽ thể hiện hệ thống xử lý tệp tin làm ví dụ theo các phương án khác nhau của sáng chế; và

Fig.4 là hình vẽ thể hiện hệ thống máy tính làm ví dụ theo các phương án khác nhau của sáng chế.

Mô tả chi tiết sáng chế

Sau đây các phương án ưu tiên của sáng chế sẽ được mô tả chi tiết thông qua các phương án làm ví dụ của sáng chế có dựa vào các hình vẽ kèm theo. Bất cứ khi nào có thể, các số chỉ dẫn giống nhau sẽ được sử dụng trên tất cả các hình vẽ để chỉ các phần giống hoặc tương tự.

Các phương án khác nhau của sáng chế đưa ra phương pháp và hệ thống xử lý tệp tin để giảm IO thường xuyên của ổ đĩa khi đọc các tệp tin và để tiết kiệm các tài nguyên hệ thống. Theo một phương án của sáng chế, phương pháp và hệ thống xử lý tệp tin của sáng chế và/hoặc các hệ thống/chương trình thích hợp khác (như hệ điều hành) có thể chạy trên, ví dụ, máy tính hoặc thiết bị thích hợp bất kỳ có hệ thống máy tính (như thiết bị đầu cuối di động).

Fig.4 minh họa hệ thống máy tính làm ví dụ 400 có thể thực hiện phương pháp và hệ thống xử lý tệp tin của sáng chế. Như được thể hiện trên Fig.4, hệ thống máy tính 400 có thể bao gồm bộ xử lý 402, vật ghi 404, màn hình 406, môđun liệu lạc 408, cơ sở dữ liệu 410, và các thiết bị ngoại vi 412. Có thể bỏ qua các thiết bị cụ thể và cũng có thể bao gồm các thiết bị khác.

Bộ xử lý 402 có thể bao gồm bộ xử lý hoặc các bộ xử lý thích hợp bất kỳ. Ngoài ra, bộ xử lý 402 có thể bao gồm nhiều lõi để xử lý đa luồng hoặc song song. Vật ghi 404 có thể bao gồm các môđun nhớ, như ROM, RAM, và các môđun nhớ nhanh, và các bộ lưu trữ lớn, như CD-ROM, ổ đĩa U, ổ đĩa cứng, v.v.. Vật ghi 404 có thể lưu các chương trình máy tính thích hợp bất kỳ để tiến hành các xử lý khác nhau, khi được tiến hành bởi bộ xử lý 402.

Ngoài ra, các thiết bị ngoại vi 412 có thể bao gồm các thiết bị I/O như bàn phím và chuột, và môđun truyền thông 408 có thể bao gồm các thiết bị mạng để thiết lập các kết nối thông qua mạng truyền thông không dây hoặc hữu tuyến. Cơ sở dữ liệu 410 có thể bao gồm một hoặc nhiều cơ sở dữ liệu để lưu trữ dữ liệu nhất định và để thực hiện các thao tác nhất định trên dữ liệu được lưu trữ, như truy vấn cơ sở dữ liệu.

Theo các phương án khác nhau, hệ thống máy tính được mô tả trên Fig.4 có thể được sử dụng hoặc được làm thích ứng để thực hiện phương pháp và hệ thống xử lý tệp tin cùng với các hệ thống/chương trình thích hợp khác. Ví dụ, theo một phương án, một cơ chế xử lý ngoại lệ có thể được sử dụng trong phương pháp/hệ thống xử lý tệp tin nêu trên để lưu trữ dữ liệu tệp tin trong vùng bộ nhớ để đạt được việc đọc tệp tin nhanh.

Như được sử dụng ở đây, cơ chế xử lý ngoại lệ được hiểu là một cơ chế bao gồm, ví dụ, định trước hàm xử lý ngoại lệ truy cập bộ nhớ và đăng ký hàm xử lý ngoại lệ truy cập bộ nhớ định trước trong một hệ điều hành. Khi việc truy cập vào bộ nhớ bởi người sử dụng là bất thường, thì hệ điều hành không cho phép hệ thống xử lý theo cách thông thường (ví dụ, tạo ra thông tin cảnh báo, v.v..). Thay vào đó, hệ điều hành có thể gọi hàm xử lý ngoại lệ để xử lý sự bất bình thường đó. Ví dụ, khi mở tệp tin, thì vùng bộ nhớ tương ứng được yêu cầu. Khi đọc dữ liệu tệp tin, thì

dữ liệu tương ứng được đọc từ vùng bộ nhớ. Nếu không tồn tại dữ liệu bộ nhớ, thì có thể xảy ra ngoại lệ truy cập bộ nhớ (ví dụ, mất trang), tức là dữ liệu tệp tin được đọc trong lần thứ nhất. Hệ thống xử lý tệp tin có thể gọi hàm xử lý ngoại lệ, và hàm xử lý ngoại lệ có thể gọi hàm SetFilePointer và hàm ReadFile để lưu trữ dữ liệu tệp tin trong vùng bộ nhớ. Nếu tồn tại dữ liệu bộ nhớ, thì đồng nghĩa với việc dữ liệu tệp tin đã được đọc một lần, sau đó có thể đọc dữ liệu tệp tin trực tiếp từ vùng bộ nhớ.

Fig.1 thể hiện phương pháp đọc tệp tin làm ví dụ có hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký trong một hệ điều hành.

Như được thể hiện trên Fig.1, trong khối 101, khi nhận yêu cầu mở tệp tin cần đọc, thì hệ thống xử lý tệp tin có thể áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở. Hệ thống xử lý tệp tin cũng có thể thiết lập thuộc tính vùng bộ nhớ là không đọc được và không ghi được. Vùng bộ nhớ có kích thước không nhỏ hơn tệp tin được yêu cầu mở. Theo một phương án, kích thước của vùng bộ nhớ bằng với kích thước của tệp tin.

Theo các phương án khác, hệ thống xử lý tệp tin có thể áp dụng cho vùng bộ nhớ bằng cách gọi hàm VirtualAlloc. Hàm VirtualAlloc là hàm API Windows và được chứa trong tệp tin Kernel32.dll của hệ Windows. Hàm VirtualAlloc được tạo ra để gọi khoảng không xử lý địa chỉ ảo và để lập biểu hoặc trình một phần của (các) trang. Hàm VirtualAlloc được sử dụng để phân bổ bộ nhớ. Hàm VirtualAlloc có thể sử dụng nhiều cuộc gọi đồng thời để trình một phần hoặc toàn bộ vùng và để giữ lại một vùng bộ nhớ lớn. Việc trình nhiều cuộc gọi của cùng một vùng không lỗi, cho phép ứng dụng, sau khi dự phòng bộ nhớ, để tự do trình trang này lên để ghi.

Trong khối 102, khi nhận được yêu cầu đọc tệp tin, thì có thể truy cập vùng bộ nhớ tương ứng với tệp tin được yêu cầu đọc. Trong khối 103, nếu việc truy cập vào bộ nhớ có lỗi và nguyên nhân lỗi là do thuộc tính vùng bộ nhớ là không đọc được và không ghi được, thì khối 104 có thể được thực hiện. Theo cách khác, nếu việc truy cập bộ nhớ không lỗi hoặc nguyên nhân lỗi không phải là thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì khối 106 có thể được chạy.

Khi người sử dụng đọc tệp tin, hệ thống xử lý tệp tin có thể đọc dữ liệu tệp tin từ vùng bộ nhớ tương ứng. Nếu đây là lần đầu tiên đọc tệp tin, thì thuộc tính của vùng bộ nhớ có thể là "không đọc được và không ghi được" như thiết lập ban đầu. Ngoại lệ truy cập bộ nhớ có thể được tạo ra và hàm xử lý ngoại lệ truy cập bộ nhớ có thể được gọi. Sau đó, khối 104 có thể được thực hiện. Trong trường hợp này, tệp tin được lưu trữ trên ổ đĩa và vẫn chưa được lưu vào vùng bộ nhớ.

Nếu người sử dụng trước đó đã đọc tệp tin (tức là không phải lần đọc tệp tin thứ nhất) và thuộc tính của vùng bộ nhớ đã được biến đổi thành "có thể đọc và có thể ghi", thì ngoại lệ truy cập bộ nhớ có thể được tạo ra. Trong trường hợp này, dữ liệu tệp tin đã được lưu trữ trong vùng bộ nhớ. Sau đó, khối 106 có thể được thực hiện trực tiếp.

Trong khối 104, hệ thống xử lý tệp tin có thể gọi hàm xử lý ngoại lệ truy cập bộ nhớ để thu từ ổ đĩa dữ liệu tệp tin yêu cầu đọc. Hàm xử lý ngoại lệ truy cập bộ nhớ có thể bao gồm hàm con trỏ thiết lập tệp tin (SetFilePointer) và hàm đọc tệp tin (ReadFile). Việc gọi hàm xử lý ngoại lệ truy cập bộ nhớ để đọc dữ liệu tệp tin từ ổ đĩa có thể tạo ra các hoạt động IO của ổ đĩa. Fig.2 thể hiện phương pháp làm ví dụ về gọi hàm xử lý ngoại lệ truy cập bộ nhớ để đọc dữ liệu tệp tin từ ổ đĩa theo các phương án khác nhau của sáng chế.

Trong khối 21, hệ thống xử lý tệp tin có thể gọi và thực hiện hàm SetFilePointer để định vị tệp tin trong ổ đĩa. Cụ thể là, các con trỏ có thể ghi nhận vị trí của tệp tin, tức là hàm con trỏ có thể định vị tệp tin vì trên đó cung từ, rãnh ghi, cụm, và byte của ổ đĩa để tìm ra một cách chính xác vị trí lưu trữ của tệp tin trên ổ đĩa.

Trong khối 22, hệ thống xử lý tệp tin có thể gọi và thực hiện hàm ReadFile để đọc dữ liệu tệp tin theo vị trí của tệp tin trên ổ đĩa. Ví dụ, khi hàm SetFilePointer hướng tới vị trí của tệp tin trên ổ đĩa, thì hệ thống xử lý tệp tin có thể ghi thông tin vị trí của tệp tin hiện thời và đọc dữ liệu tệp tin bằng cách gọi hàm ReadFile. Hàm ReadFile thường có bốn tham số được sử dụng để lần lượt biểu thị: tệp tin cần đọc, vùng đệm để lưu dữ liệu tệp tin đã đọc, độ dài của tệp tin cần đọc, và kích thước thực sự của tiếp tin đã đọc. Ví dụ, hàm SetFilePointer có thể định vị tệp tin trên ổ

đĩa trên cung từ thứ ba, rãnh ghi thứ hai, cụm thứ hai, và byte thứ ba. Nghĩa là, khi hàm ReadFile được gọi để đọc tệp tin, thì tệp tin có thể được đọc bắt đầu từ byte thứ ba và dữ liệu có độ dài xác định có thể được đặt vào trong vùng đệm xác định.

Theo một phương án, yêu cầu đọc tệp tin có thể là yêu cầu đọc dữ liệu tệp tin từng phần. Bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ, dữ liệu tệp tin thu được từ ổ đĩa có thể là dữ liệu tệp tin từng phần được yêu cầu đọc. Cụ thể là, khi nhận yêu cầu của người sử dụng đọc dữ liệu trong vùng nhất định của tệp tin (ví dụ, để đọc duy nhất một trạng cụ thể của dữ liệu tệp tin), thì hệ thống xử lý tệp tin có thể không truy cập vào toàn bộ vùng bộ nhớ, mà truy cập vào vùng bộ nhớ nhất định tương ứng với dữ liệu trong vùng nhất định của tệp tin (ví dụ, vùng bộ nhớ tương ứng với dữ liệu tệp tin của trang cụ thể đó).

Nếu thuộc tính vùng bộ nhớ tương ứng với dữ liệu trong vùng nhất định của tệp tin, ví dụ, dữ liệu tệp tin của trang cụ thể đó, là không đọc được và không ghi được, thì dữ liệu tệp tin của trang cụ thể đó có thể thu được từ ổ đĩa bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ (ví dụ, bao gồm hàm SetFilePointer và hàm ReadFile). Dữ liệu tệp tin của trang cụ thể đó có thể được lưu trữ trong vùng bộ nhớ tương ứng. Thuộc tính vùng bộ nhớ tương ứng có thể được thiết lập có thể đọc và có thể ghi. Dữ liệu tệp tin của trang cụ thể đó đọc được từ vùng bộ nhớ tương ứng. Nếu vùng bộ nhớ tương ứng có thuộc tính là đọc được và ghi được, đồng nghĩa với việc dữ liệu tệp tin của trang cụ thể đó lần đầu không được đọc, thì dữ liệu tệp tin của trang cụ thể đó có thể đọc trực tiếp từ vùng bộ nhớ tương ứng.

Theo các phương án khác, dữ liệu tệp tin từng phần mà cần đọc có thể có thể được xác định là cần thiết, ví dụ, bởi người sử dụng, trong hàm ReadFile. Dữ liệu tệp tin từng phần có thể được lưu trữ trong vùng bộ nhớ mà không phải đọc toàn bộ dữ liệu tệp tin. Việc sử dụng quá mức vùng bộ nhớ và lãng phí một cách không cần thiết các tài nguyên của hệ thống có thể tránh được.

Trong khối 105, hệ thống xử lý tệp tin có thể lưu trữ dữ liệu tệp tin trong vùng bộ nhớ tương ứng và có thể biến đổi thuộc tính của vùng bộ nhớ để đọc được và ghi được. Sau khi hệ thống xử lý tệp tin kết thúc thao tác lưu trữ dữ liệu tệp tin trong vùng bộ nhớ, thì thuộc tính của vùng bộ nhớ có thể được biến đổi thành đọc

được và ghi được, mà ban đầu được thiết lập là không đọc được và không ghi được. Khi tệp tin một lần nữa được đọc, thì tệp tin có thể đọc trực tiếp từ vùng bộ nhớ và các hoạt động IO của ổ đĩa có thể tránh được.

Trong khối 106, hệ thống xử lý tệp tin có thể đọc dữ liệu tệp tin từ vùng bộ nhớ. Trong khối 107, khi nhận yêu cầu đóng tệp tin, thì hệ thống xử lý tệp tin có thể giải phóng vùng bộ nhớ tương ứng với tệp tin được yêu cầu đóng.

Theo cách này, phương pháp và hệ thống xử lý tệp tin của sáng chế có thể bao gồm, ví dụ, đăng ký hàm xử lý ngoại lệ truy cập bộ nhớ trong một hệ điều hành, áp dụng cho vùng bộ nhớ tương ứng với tệp tin cần được mở, và đọc dữ liệu tệp tin từ vùng bộ nhớ. Nếu việc truy cập vào bộ nhớ bị lỗi và nguyên nhân lỗi là do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì có thể thu được dữ liệu tệp tin được yêu cầu đọc từ ổ đĩa bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ.

Như được mô tả ở trên, khi dữ liệu tệp tin từ một vùng cụ thể được đọc, thì có thể tạo ra các hoạt động IO của ổ đĩa chỉ khi dữ liệu tệp tin được đọc trong lần thứ nhất. Sau đó, dữ liệu tệp tin có thể được lưu trữ trong vùng bộ nhớ. Nếu sau đó dữ liệu tệp tin cần đọc một lần nữa từ cùng một vùng nhất định, thì có thể đọc trực tiếp dữ liệu tệp tin từ vùng bộ nhớ. Không IO nào của ổ đĩa có thể được tạo ra một lần nữa. Các tài nguyên của hệ thống có thể được tiết kiệm để cải thiện một cách hiệu quả việc đọc các tệp tin.

Fig.3 thể hiện hệ thống xử lý tệp tin có hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký trong một hệ điều hành. Như được thể hiện trên Fig.3, một hệ thống làm ví dụ có thể bao gồm môđun nhận 31, môđun xử lý thứ nhất 32, môđun xử lý thứ hai 33, và/hoặc môđun thu 34.

Môđun nhận 31 có thể được tạo cấu hình để nhận yêu cầu mở tệp tin và để nhận yêu cầu đọc tệp tin. Khi môđun nhận 31 nhận yêu cầu mở tệp tin, thì môđun xử lý thứ nhất 32 có thể được tạo cấu hình để áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở và để thiết lập thuộc tính vùng bộ nhớ là không đọc được và không ghi được.

Khi môđun nhận 31 nhận yêu cầu đọc tệp tin, thì môđun xử lý thứ hai 33 có thể được tạo cấu hình để truy cập vùng bộ nhớ tương ứng với tệp tin được yêu cầu đọc. Nếu việc truy cập vào bộ nhớ có lỗi và nguyên nhân lỗi là do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì có thể thu được dữ liệu tệp tin được yêu cầu đọc từ ổ đĩa bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ. Sau đó, dữ liệu tệp tin thu được có thể được lưu trữ trong vùng bộ nhớ và thuộc tính vùng bộ nhớ có thể được thiết lập là đọc được và ghi được. Dữ liệu tệp tin đọc được từ vùng bộ nhớ bằng cách gọi môđun thu 34. Theo cách khác, nếu việc truy cập bộ nhớ không có lỗi hoặc nguyên nhân có lỗi không phải là thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì môđun thu 34 có thể được gọi để đọc dữ liệu tệp tin trực tiếp từ vùng bộ nhớ.

Môđun thu 34 có thể đọc dữ liệu tệp tin từ vùng bộ nhớ. Cụ thể là, hàm xử lý ngoại lệ truy cập bộ nhớ có thể bao gồm các hàm SetFilePointer và hàm ReadFile. Môđun xử lý thứ hai 33 có thể được sử dụng để định vị tệp tin trên ổ đĩa bằng cách gọi và chạy hàm SetFilePointer; và để đọc dữ liệu tệp tin theo vị trí của tệp tin trên ổ đĩa bằng cách gọi và chạy hàm ReadFile.

Theo một phương án, yêu cầu đọc tệp tin có thể là yêu cầu đọc dữ liệu tệp tin từng phần. Môđun xử lý thứ hai 33 có thể được sử dụng để gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ để thu dữ liệu tệp tin từng phần được yêu cầu đọc từ ổ đĩa.

Theo một phương án, khi môđun nhận 31 nhận yêu cầu mở tệp tin, thì môđun xử lý thứ nhất 32 có thể được sử dụng để áp dụng cho vùng bộ nhớ có kích thước không nhỏ hơn kích thước của tệp tin, và để thiết lập mối quan hệ tương ứng giữa vùng bộ nhớ và tệp tin.

Tùy ý, hệ thống xử lý tệp tin có thể còn bao gồm môđun xử lý thứ ba 35. Trong trường hợp này, môđun nhận 31 có thể được sử dụng để nhận yêu cầu đóng tệp tin. Khi môđun nhận 31 nhận yêu cầu đóng tệp tin, thì môđun xử lý thứ ba 35 có thể được sử dụng để giải phóng vùng bộ nhớ tương ứng với tệp tin được yêu cầu đóng.

Theo các phương án khác, các môđun đã nêu có thể được tạo cấu hình trong một thiết bị hoặc được tạo cấu hình trong nhiều thiết bị nếu muốn. Các môđun

được đề cập ở đây có thể được tích hợp trong một môđun hoặc trong nhiều môđun. Mỗi trong số các môđun được đề cập ở đây có thể được chia thành một hoặc nhiều môđun con, mà có thể được kết hợp theo cách bất kỳ.

Trong các phương án được mô tả (ví dụ, như được thể hiện trên các hình vẽ từ Fig.1 đến Fig.4) chỉ là các phương án làm ví dụ. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể hiểu rõ rằng các phần mềm và/hoặc các phần cứng thích hợp (ví dụ, nền tảng phần cứng vạn năng) có thể được bao gồm và được sử dụng theo các phương án khác nhau của sáng chế. Ví dụ, các phương án nêu trên có thể được thực hiện bởi chỉ một phần cứng, mà theo cách khác có thể chỉ thực hiện được bởi các sản phẩm phần mềm. Các sản phẩm phần mềm này có thể được lưu trữ trong vật ghi. Các sản phẩm phần mềm này có thể bao gồm các lệnh thích hợp để cho phép thiết bị đầu cuối (ví dụ, bao gồm điện thoại di động, máy tính cá nhân, máy chủ hoặc thiết bị mạng, v.v..) thực hiện các phương án được bộc lộ.

Các ứng dụng, ưu điểm, các phương án thay thế, các biến thể hoặc các phương án tương đương với các phương án của sáng chế là có thể tạo ra được bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật.

Khả năng áp dụng công nghiệp và hiệu quả của sáng chế

Khả năng áp dụng công nghiệp và hiệu quả của sáng chế theo các phương án được mô tả trên đây được đưa ra chỉ nhằm minh họa sáng chế mà không giới hạn phạm vi của sáng chế. Các phương án khác, các biến thể hoặc các phương án tương đương với giải pháp kỹ thuật được mô tả trong các phương án nêu trên của sáng chế là thể tạo ra được bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật và thuộc phạm vi của sáng chế.

Phương pháp và hệ thống xử lý tệp tin của sáng chế có thể bao gồm, ví dụ, đăng ký hàm xử lý ngoại lệ truy cập bộ nhớ trong một hệ điều hành, áp dụng cho vùng bộ nhớ tương ứng với tệp tin cần được mở, và đọc dữ liệu tệp tin từ vùng bộ nhớ. Nếu việc truy cập vào bộ nhớ có lỗi và nguyên nhân lỗi là do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì có thể thu được dữ liệu tệp tin được yêu cầu đọc từ ổ đĩa bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ. Như được mô tả ở trên, khi dữ liệu tệp tin từ một vùng cụ thể được đọc, thì có

thể tạo ra các hoạt động IO của ổ đĩa chỉ khi dữ liệu tệp tin được đọc trong lần thứ nh. Sau đó, dữ liệu tệp tin có thể được lưu trữ trong vùng bộ nhớ. Nếu sau khi dữ liệu tệp tin cần đọc một lần nữa từ cùng một vùng nhất định, thì có thể đọc trực tiếp dữ liệu tệp tin từ vùng bộ nhớ. Không IO nào của ổ đĩa có thể được tạo ra một lần nữa. Các tài nguyên của hệ thống có thể được tiết kiệm để cải thiện một cách hiệu quả việc đọc các tệp tin.

Yêu cầu bảo hộ

1. Phương pháp xử lý tệp tin dựa vào hệ điều hành, phương pháp này bao gồm các bước:

nhận yêu cầu mở tệp tin; tệp tin này được lưu trữ tại ổ đĩa;

áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở;

đáp lại yêu cầu mở tệp tin, khi vùng bộ nhớ được áp dụng được gán cho tệp tin, thiết lập thuộc tính của vùng bộ nhớ là không đọc được và không ghi được;

nhận yêu cầu đọc tệp tin;

đáp lại yêu cầu đọc tệp tin, truy cập vùng bộ nhớ tương ứng với tệp tin;

khi việc truy cập vùng bộ nhớ để đọc tệp tin có lỗi do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì gọi hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký với hệ điều hành để khởi tạo các hoạt động sau:

thu dữ liệu tệp tin được yêu cầu để đọc từ ổ đĩa,

lưu trữ dữ liệu tệp tin thu được trong vùng bộ nhớ, và thiết lập thuộc tính của vùng bộ nhớ là đọc được và ghi được; và

đọc dữ liệu tệp tin từ vùng bộ nhớ.

2. Phương pháp theo điểm 1, trong đó hàm xử lý ngoại lệ truy cập bộ nhớ bao gồm hàm con trỏ thiết lập tệp tin (SetFilePointer) và hàm đọc tệp tin (ReadFile), và trong đó bước thu dữ liệu tệp tin bằng cách gọi hàm xử lý ngoại lệ truy cập bộ nhớ bao gồm:

định vị tệp tin trên ổ đĩa bằng cách gọi và chạy hàm SetFilePointer; và

đọc dữ liệu tệp tin theo tệp tin đã định vị trên ổ đĩa bằng cách gọi và chạy hàm ReadFile.

3. Phương pháp theo điểm 1, trong đó yêu cầu đọc tệp tin là yêu cầu đọc dữ liệu tệp tin từng phần và trong đó, bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ, dữ liệu tệp tin thu được từ ổ đĩa là dữ liệu tệp tin từng phần được yêu cầu đọc.

4. Phương pháp theo điểm 1, trong đó bước áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở bao gồm:

áp dụng cho vùng bộ nhớ có kích thước không nhỏ hơn tệp tin; và
thiết lập mối quan hệ tương ứng giữa vùng bộ nhớ và tệp tin.

5. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước:

nhận yêu cầu đóng tệp tin, và
giải phóng vùng bộ nhớ.

6. Phương pháp theo điểm 1, trong đó khi việc truy cập vào vùng bộ nhớ của hệ điều hành là bất thường, hệ điều hành kích hoạt hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký thay vì tuân theo thủ tục xử lý thông thường của hệ thống cho hoạt động bất thường.

7. Phương pháp theo điểm 3, trong đó bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ, thuộc tính của vùng bộ nhớ từng phần tương ứng với dữ liệu tệp tin từng phần thu được từ ổ đĩa được thiết lập là đọc được và ghi được; và thuộc tính của vùng bộ nhớ còn lại ngoài vùng bộ nhớ từng phần là không đổi.

8. Phương pháp theo điểm 1, trong đó:

sau khi tệp tin được mở, dữ liệu của tệp tin là dữ liệu chỉ đọc từ ổ đĩa một lần khi dữ liệu được yêu cầu lần đầu tiên, và sau lần đầu tiên, dữ liệu của tệp tin được đọc từ vùng bộ nhớ được gán.

9. Phương pháp theo điểm 1, trong đó bước áp dụng cho vùng bộ nhớ còn bao gồm:

gọi hàm VirtualAlloc trong hệ thống window, hàm VirtualAlloc được tạo cấu hình để sử dụng nhiều lời gọi song song để cấp phát bộ nhớ; và
cấp phát vùng bộ nhớ tương ứng với tệp tin bởi hàm VirtualAlloc.

10. Hệ thống xử lý tệp tin dựa vào hệ điều hành, hệ thống này bao gồm ít nhất một bộ xử lý, ổ đĩa, bộ nhớ, và nhiều môđun chương trình được thực thi bởi bộ xử lý, các môđun chương trình bao gồm:

môđun nhận được tạo cấu hình để nhận yêu cầu mở tệp tin và để nhận yêu cầu đọc tệp tin, tệp tin này được lưu trữ trong tại ổ đĩa;

môđun xử lý thứ nhất được tạo cấu hình, khi môđun nhận nhận yêu cầu mở tệp tin, để áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở; và đáp lại yêu cầu mở tệp tin, khi vùng bộ nhớ được áp dụng được gán cho tệp tin, thiết lập thuộc tính của vùng bộ nhớ là không đọc được và không ghi được;

môđun xử lý thứ hai được tạo cấu hình, khi môđun nhận nhận yêu cầu đọc tệp tin, để truy cập vùng bộ nhớ tương ứng với tệp tin; và nếu việc truy cập vào vùng bộ nhớ để đọc tệp tin có lỗi do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, gọi hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký với hệ điều hành để khởi tạo các hoạt động sau: gọi dữ liệu tệp tin được yêu cầu để đọc từ ổ đĩa, lưu trữ dữ liệu tệp tin thu được trong vùng bộ nhớ, và thiết lập thuộc tính của vùng bộ nhớ là đọc được và ghi được; và

môđun thu được tạo cấu hình để đọc dữ liệu tệp tin từ vùng bộ nhớ.

11. Hệ thống theo điểm 10, trong đó hàm xử lý ngoại lệ truy cập bộ nhớ bao gồm hàm con trỏ thiết lập tệp tin (SetFilePointer) và hàm đọc tệp tin (ReadFile), và trong đó môđun xử lý thứ hai được tạo cấu hình để:

định vị tệp tin trên ổ đĩa bằng cách gọi và chạy hàm SetFilePointer; và

đọc dữ liệu tệp tin theo tệp tin đã định vị trên ổ đĩa bằng cách gọi và chạy hàm ReadFile.

12. Hệ thống theo điểm 10, trong đó yêu cầu đọc tệp tin là yêu cầu đọc dữ liệu tệp tin từng phần và trong đó môđun xử lý thứ hai được tạo cấu hình sao cho, bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ, dữ liệu tệp tin thu được từ ổ đĩa là dữ liệu tệp tin từng phần được yêu cầu đọc.

13. Hệ thống theo điểm 10, trong đó môđun xử lý thứ nhất được tạo cấu hình, khi môđun nhận nhận yêu cầu mở tệp tin, để:

áp dụng cho vùng bộ nhớ có kích thước không nhỏ hơn tệp tin; và

thiết lập mối quan hệ tương ứng giữa vùng bộ nhớ và tệp tin.

14. Hệ thống theo điểm 10, trong đó hệ thống này còn bao gồm môđun xử lý thứ ba, trong đó môđun nhận được tạo cấu hình để nhận yêu cầu đóng tệp tin và môđun xử lý thứ ba được tạo cấu hình để giải phóng vùng bộ nhớ.

15. Vật ghi không khả biến đọc được bằng máy tính có chương trình máy tính để, khi được thực thi bởi bộ xử lý, thực hiện phương pháp xử lý tệp tin dựa vào hệ điều hành, phương pháp này bao gồm các bước:

nhận yêu cầu mở tệp tin; tệp tin này được lưu trữ tại ổ đĩa;

áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở;

đáp lại yêu cầu mở tệp tin, khi vùng bộ nhớ được áp dụng được gán cho tệp tin, thiết lập thuộc tính của vùng bộ nhớ là không đọc được và không ghi được;

nhận yêu cầu đọc tệp tin;

đáp lại yêu cầu đọc tệp tin, truy cập vùng bộ nhớ tương ứng với tệp tin;

khi việc truy cập vùng bộ nhớ để đọc tệp tin có lỗi do thuộc tính của vùng bộ nhớ là không đọc được và không ghi được, thì gọi hàm xử lý ngoại lệ truy cập bộ nhớ được đăng ký với hệ điều hành để khởi tạo các hoạt động sau:

thu dữ liệu tệp tin được yêu cầu để đọc từ ổ đĩa,

lưu trữ dữ liệu tệp tin thu được trong vùng bộ nhớ, và thiết lập thuộc tính của vùng bộ nhớ là đọc được và ghi được; và

đọc dữ liệu tệp tin từ vùng bộ nhớ.

16. Vật ghi theo điểm 15, trong đó trong đó hàm xử lý ngoại lệ truy cập bộ nhớ bao gồm hàm con trỏ thiết lập tệp tin (SetFilePointer) và hàm đọc tệp tin (ReadFile), và trong đó bước thu dữ liệu tệp tin bằng cách gọi hàm xử lý ngoại lệ truy cập bộ nhớ bao gồm:

định vị tệp tin trên ổ đĩa bằng cách gọi và chạy hàm SetFilePointer; và

đọc dữ liệu tệp tin theo tệp tin đã định vị trên ổ đĩa bằng cách gọi và chạy hàm ReadFile.

17. Vật ghi theo điểm 15, trong đó yêu cầu đọc tệp tin là yêu cầu đọc dữ liệu tệp tin từng phần và trong đó, bằng cách gọi và chạy hàm xử lý ngoại lệ truy cập bộ nhớ, dữ liệu tệp tin thu được từ ổ đĩa là dữ liệu tệp tin từng phần được yêu cầu đọc.

18. Vật ghi theo điểm 15, trong đó bước áp dụng cho vùng bộ nhớ tương ứng với tệp tin được yêu cầu mở bao gồm:

áp dụng cho vùng bộ nhớ có kích thước không nhỏ hơn tệp tin; và

thiết lập mối quan hệ tương ứng giữa vùng bộ nhớ và tệp tin.

19. Vật ghi theo điểm 15, trong đó phương pháp này còn bao gồm bước:

nhận yêu cầu đóng tệp tin, và

giải phóng vùng bộ nhớ.

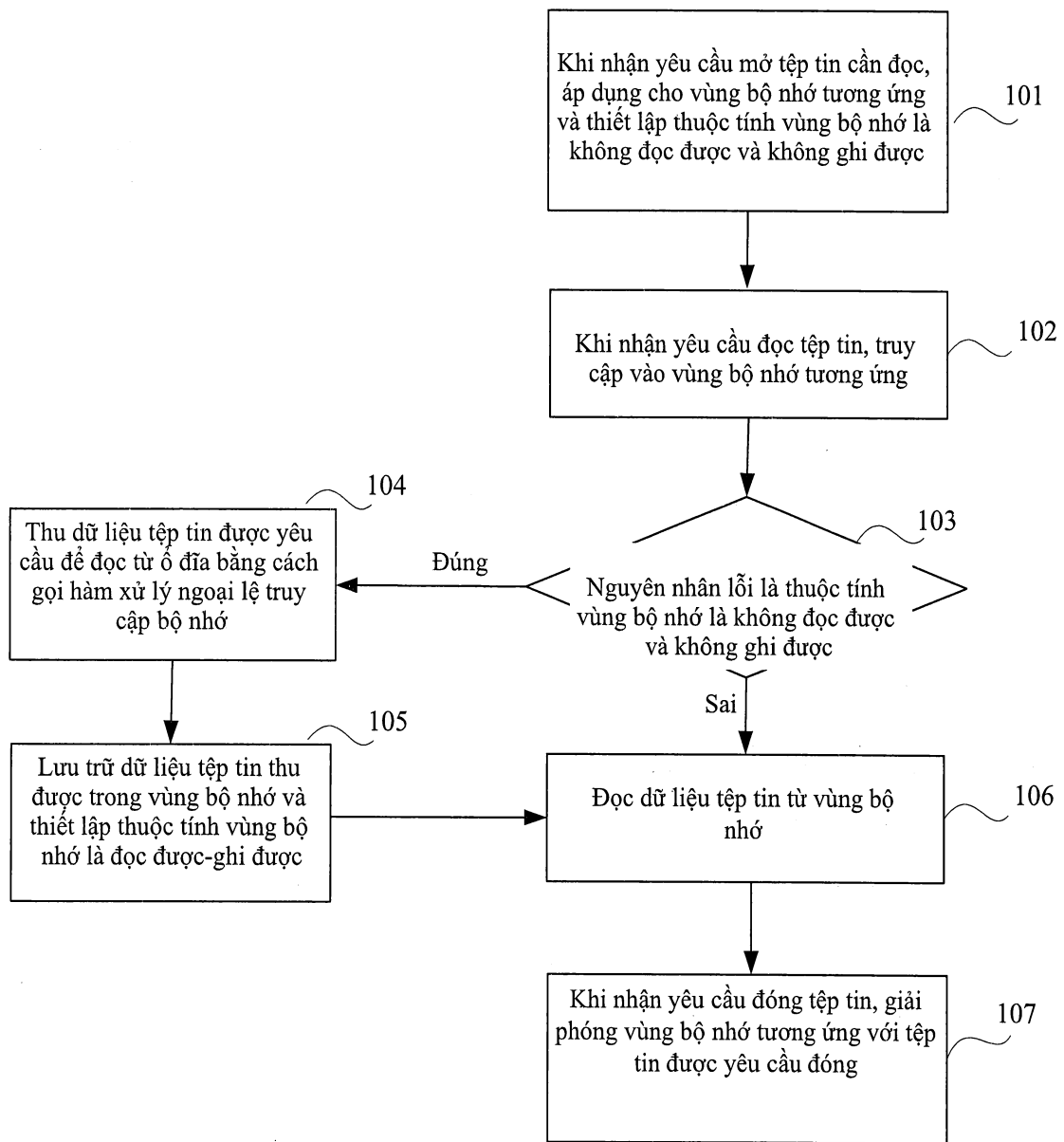


FIG. 1

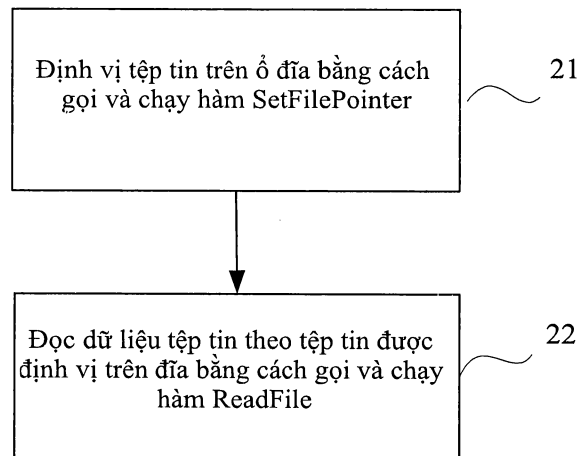


FIG. 2

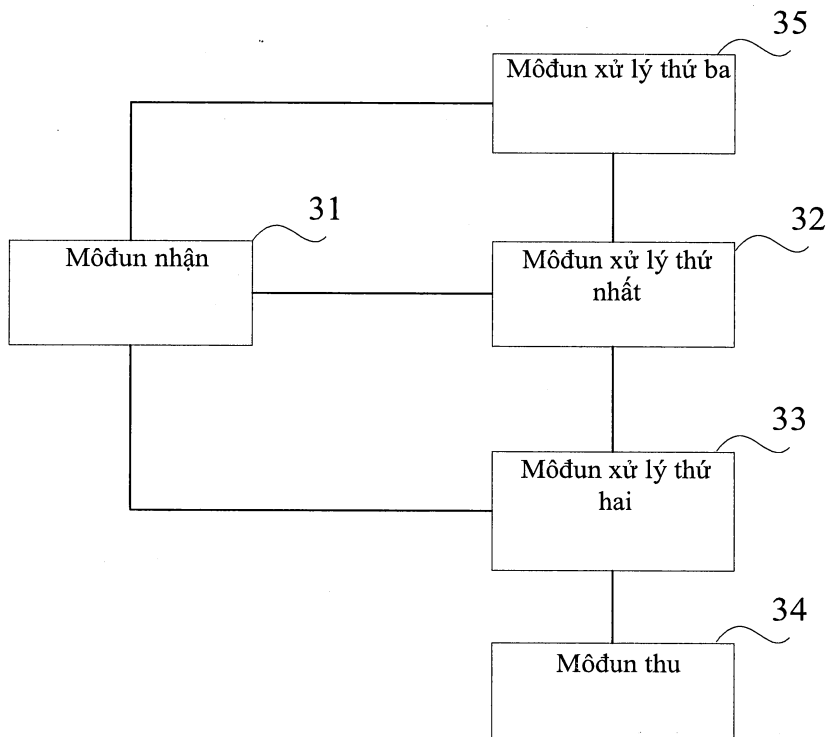


FIG. 3

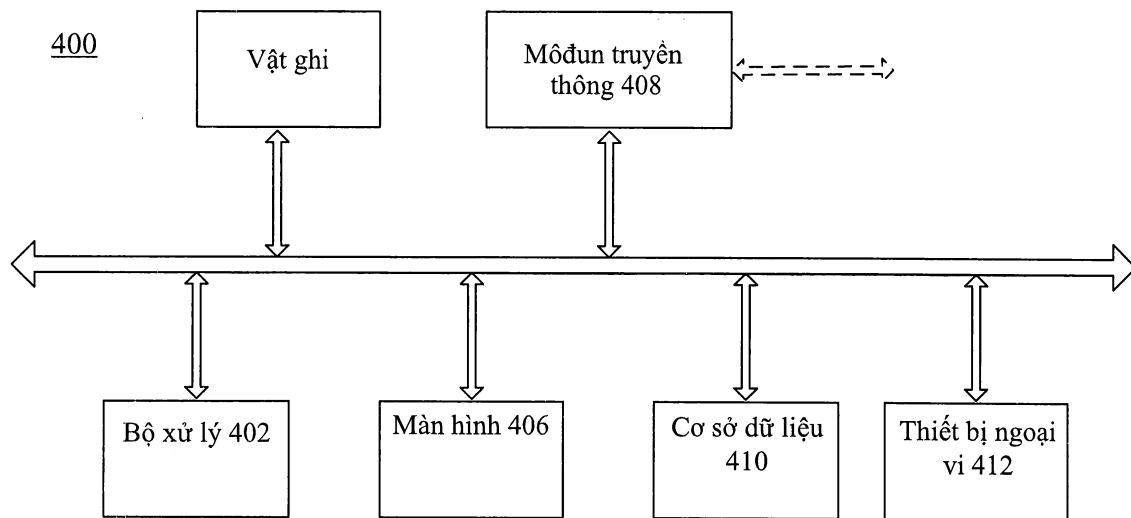


FIG. 4