

Đại học quốc gia thành phố Hồ Chí Minh

Trường Đại học Khoa Học Tự Nhiên



Hệ điều hành

Đồ án 2: Xử lý đa chương cho hệ điều hành nachos

Nhóm thực hiện:

Nguyễn Văn Phước	1612523
Châu Hoàng Phúc	1612520
Nguyễn Thị Thu Quyền	1612548
Nguyễn Trương Quang	1612533

Giáo viên:

Phạm Tuấn Sơn

MỤC LỤC

I. LỚP THREADMANAGE	3
1. CHỨC NĂNG:	3
2. CÀI ĐẶT:.....	3
II. SYSCALL EXEC:	3
1. CHỨC NĂNG:	3
2. CÁC CÀI ĐẶT:	4
III.LỚP PAGETABLEMANAGE:	6
1. CHỨC NĂNG:	6
2. CÀI ĐẶT:.....	6
IV.XỬ LÝ ĐA CHƯƠNG CHO NACHOS:	6
1. CHỈNH SỬA VIỆC CẤP PHÁT BẢNG TRANG CHO TIỀN TRÌNH:	6
2. CHỈNH SỬA VIỆC NẠP CHƯƠNG TRÌNH LÊN BỘ NHỚ CHÍNH:.....	7
3. TĂNG SỐ LƯỢNG PAGE CHO MÁY ẢO NACHOS:.....	7

I. Lớp ThreadManage

1. Chức năng:

- Có nhiệm vụ quản lý các tiến trình của nachos dưới dạng là các tiến trình của linux. Cho phép quản lý tối đa 10 tiến trình chạy một lúc. Có các phương thức tìm vị trí trống, thêm tiến trình, xóa tiến trình.

2. Cài đặt:

- Lớp sẽ lưu trữ một mảng con trỏ các tiến trình và kích thước của mảng.
- Gồm các phương thức sau:
 - **FindFreeSlot** tìm vị trí trống để lưu thông tin của một tiến trình. Trả về -1 nếu không còn vị trí trống ngược lại trả về vị trí trống.
 - **Add** thêm tiến trình vào mảng tiến trình. Trả về true nếu thành công, ngược lại trả về false.
 - **GetThread** lấy tiến trình theo id là vị trí của tiến trình. Trả về con trỏ của tiến trình nếu thành công, ngược lại trả về null.
 - **EndThread** xóa tiến trình ra khỏi mảng tiến trình. Trả về true nếu thành công, ngược lại trả về false.

II. Syscall Exec:

1. Chức năng:

- Syscall hỗ trợ thực thi một chương trình trong chương trình hiện hành. Tham số nhận vào là tên của chương trình.
- Trả về id của chương trình là vị trí trong mảng tiến trình của Class **ThreadManage** nếu thành công. Nếu thất bại trả về -1.

2. Các cài đặt:

- Syscall Exec đã được định nghĩa trong các tập tin **syscall.h** và được ánh xạ **start.c** và **start.s**
- Ta cần vào file exception.cc và xử lý với case type là SC_Exec. Ta sẽ định nghĩa các công việc thực hiện của syscall như sau:
 - Nhận tên file thực thi từ thanh ghi \$4.
 - Chuyển tên file từ vùng userspace sang kernel space thông qua hàm User2System.
 - Tạo mới một tiểu trình với tên là tên của chương trình thực thi.
 - Bắt đầu tiểu trình bằng cách set trạng thái của tiểu trình thành RUNNING bằng hàm setStatus của lớp Thread.
 - Tìm vị trí trong cho tiểu trình trong mảng tiểu trình. Thông qua hàm FindFreeSlot của lớp ThreadManage.
 - Nếu còn vị trí trống ta thêm tiểu trình vào mảng thông qua hàm Add của lớp ThreadManage. Nếu không còn ta kết thúc syscall và trả về -1 vào thanh ghi \$2.
 - Trường hợp còn vị trí trống ta dùng hàm **Fork** của lớp Thread để chạy hàm **StartProcess_2** với tham số là **id** của tiểu trình.
 - Cuối cùng ta kết thúc syscall và trả về vị trí của tiểu trình trong mảng tiểu trình. Gán vào thanh ghi \$2.

- Hàm **StartProcess_2**:
 - Hàm sẽ nhận tham số là id của tiểu trình cần thực thi. Và lấy tên của tiểu trình từ mảng tiểu trình.
 - Nếu lấy thành công tên file. Chúng ta sẽ thực hiện mở file thực thi bằng hàm **Open** của lớp **fileSystem** là lớp xử lý file của hệ điều hành. Nếu không lấy được tên file thì sẽ kết thúc hàm.
 - Khi đã mở được file thực thi. Ta sẽ tạo cho file một vùng để chứa lệnh và vùng chứa dữ liệu nằm trên ram của hệ điều hành.
 - Lúc này ta có thể đóng file thực thi vì dữ liệu đã được ghi vào ram.
 - Ta lưu lại trạng thái của ram bằng các hàm **InitRegisters** và **RestoreState** của lớp **AddrSpace**
 - Cuối cùng ta gọi hàm **Run()** của **machine**. Sẽ thực thi chương trình đang nằm trên Ram.
- Giải thích về hàm Fork của lớp Thread: Hàm Fork sẽ giúp tiểu trình hiện tại thực hiện một hàm. Trong trường hợp này là hàm **StartProcess_2** và kèm theo đó là id của tiểu trình.

III. Lớp PageTableManage:

1. Chức năng:

- Quản lý việc cấp phát, thu hồi bảng trang của hệ điều hành.

2. Cài đặt:

- Lớp sử dụng một đối tượng BitMap để quản lý việc phân trang, với số trang bằng với số trang vật lý mà hệ điều hành nachos đang quản lý.
- Gồm có các phương thức hỗ trợ như sau:
 - o FindFreeSlot: Tìm vị trí bản trang trống, nếu có trả ra vị trí còn trống ngược lại thì trả về -1.
 - o Free: giải phóng vị trí bản trang tại vị trí truyền vào.
 - o Add: Đánh dấu bản trang đang được sử dụng.

IV. Xử lý đa chương cho nachos:

1. Chỉnh sửa việc cấp phát bảng trang cho tiến trình:

- Với việc cấp phát bảng trang như hiện tại thì ta không thể chạy được đa tiến trình trên nachos được nên ta cần sửa lại việc cấp phát bản trang trong file **addrspace.cc** ở hàm dựng của lớp **AddrSpace**.
- Hiện tại quá trình cấp phát của nachos là dùng số thứ tự bảng trang của bộ nhớ logic để làm số bảng trang ứng với bên bộ nhớ vật lý. Nên ta sẽ sử dụng đối tượng của lớp **PageTableManage** đã được định nghĩa ở trên để cấp phát bảng trang vật lý cho bản trang logic thông qua phương thức **FindFreeSlot** và phương thức **Add**:
 - o Nếu hàm FindFreeSlot trả về giá trị -1 thì ta sẽ báo lỗi thông qua macro **ASSERT** để dừng hệ điều hành nachos.
 - o Ngược lại ta tiến hành sử dụng hàm Add để lưu lại vị trí đã sử dụng và gán vào cho bản trang logic tương ứng.
- Đồng thời ta cũng chỉnh sửa lại hàm hủy của lớp **AddrSpace** để giải phóng các bảng trang mà tiến trình đã chiếm.

2. Chỉnh sửa việc nạp chương trình lên bộ nhớ chính:

- Hiện tại nachos sử dụng việc nạp dữ liệu lên bộ nhớ chính theo hình thức chia segment gồm phần code và phần data nên sẽ xảy ra hiện tượng internal fragmentation, nên ta sẽ chỉnh sửa lại việc nạp chương trình để hạn chế việc xảy ra hiện tượng phân mảnh.
- Ta sẽ lưu lại mảng các vị trí của trang đã được cấp cho tiến trình bằng mảng arr kiểu int.
- Đầu tiên ta vẫn sẽ thêm vùng code vào bộ nhớ chính, sau đó ta sẽ tính toán xem vùng data cuối cùng của vùng code đã lấp đầy trang được cấp chưa, nếu chưa ta sẽ tiếp tục chép vùng data với phần còn trống đó.
- Sau đó ghi phần còn lại của vùng data vào bộ nhớ chính.

3. Tăng số lượng page cho máy ảo nachos:

- Hiện tại máy ảo để chạy nachos chỉ cung cấp 32page cho vùng nhớ chính nên lúc chạy và cấp phát cho 3 tiến trình trong đồ án này thì không đủ vùng nhớ để cấp nên ta sẽ tăng số lượng page lên 50 ở file machine.h