Đại học quốc gia thành phố Hồ Chí Minh

Trường Đại học Khoa Học Tự Nhiên

Hệ điều hành

Đồ án 3: Đa chương và đồng bộ hóa trên 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 PCB

1. Chức năng:

- Lưu các thông tin để quản lý các process với các biến Semaphore hổ trợ các quá trình join, exit, truy xuất đọc quyền .

2. Cài đặt:

- Lớp sẽ lưu trữ các Semaphore cho quá trình join, exit, truy xuất đọc quyền, số lượng tiến trình đã join, exitcode của tiến trình và id của tiến trình cha.
- Gồm các phương thức sau:
    - **Exec** tạo một tiến trình mới, thực thi và trả về id của nó.
    - **GetId** trả về id của tiến trình.
    - **GetNumWait** trả về số lượng tiến trình chờ.
    - **JoinWait** khi tiến trình cha gọi sẽ thực hiện việc chờ chó tới khi tiến trình con kết thúc.
    - **ExitWait** giúp cho tiến trình con kết thúc tiến trình.
    - **JoinRelease** sau khi kết thúc, tiến trình con sẽ gọi hàm để báo cho tiến trình cha không phải chờ nữa. Và chờ tiến trình cha cùng kết thúc.
    - **ExitRelease** tiến trình cha báo cho các tiến trình con đã hoạt động xong và cùng kết thúc.
    - **IncNumWait** tăng số tiến trình chờ.
    - **DecNumWait** giảm số tiến trình chờ.
    - **SetExitCode** đặt exitcode của tiến trình.
    - **GetExitCode** lấy exitcode của tiến trình.
    - **SetFileName** đặt tên cho tiến trình.
    - **GetFileName** trả về tên tiến trình.

II. Lớp PTable:

1. Chức năng:

- Dùng để quản lý các tiến trình đang chạy, gồm một mảng các pcb tối đa là 10 phần tử.

- Hàm constructor của lớp sẽ khởi tạo tiến trình cha (nằm ở phần tử 0). Và từ tiến trình này chúng ta sẽ tạo các tiến trình con thông qua system call Exec().

2. Cài đặt:

Gồm các phương thức sau:

- ExecUpdate đưa vào tên chương trình cần thực hiện, kiểm tra tính hợp lệ của chương trình và thực hiện chương trình bằng phương thức Exec của lớp PCB. Phương thức trả về kết quả thực hiện của phương thức Exec.
- ExitUpdate xử lí việc kết thúc của các tiến trình. Nếu là tiến trình cha (vị trí 0) thì gọi Halt(). Ngược lại nếu là tiến trình con thì sẽ SetExitCode và gọi JoinRelease để thông báo cho tiến trình cha của nó tiếp tục, và gọi ExitWait để đợi tiến trình cha cùng kết thúc.
- JoinUpdate tiến trình cha sẽ gọi để chờ tiến trình con, sau khi tiến trình con thực hiện xong thì xử lí exitcode của tiến trình. Nếu đã xong sẽ gọi ExitRelease để thông báo tiến trình con cùng thoát.
- GetFreeSlot tìm vị trí trống trong mảng tiến trình.
- IsExist(int pid) tìm xem id có tồn tại không.
- Remove xóa một tiến trình.
- GetFileName lấy tên của tiến trình.

III. Lớp Sem

1. Chức năng:

- Dùng để quản lý Semaphore các thuộc tính và phương thức của semaphore.

2. Cài đặt:

- Gồm các phương thức khởi tạo và hủy Sem.
- Wait() thực hiện thao tác chờ.
- Signal() thực hiện thao tác giải phóng Semaphore.

IV. Lớp Stable

1. Chức năng:

- Là một mảng chưa các Sem để quản lý.

2. Cài đặt:

- Các phương thức:
    - Create tạo một đối tượng Sem mới với tên của Sem đó.

- Wait thực hiện kiểm tra hợp lệ, nếu hợp lệ thì gọi this->P(), không hợp lệ sẽ báo lỗi.
- Signal thực hiện kiểm tra hợp lệ, nếu hợp lệ thì gọi this->V(), không sẽ báo lỗi.
- FindFreeSlot tìm chỗ trống trong mảng.

V. Các SystemCall

1. SC_Exec

a. Chức năng:

- Sử dụng lớp PCB và Ptable để gọi thực thi một chương trình trong một system thread mới.

b. Cài đặt:

- Đọc tên chương trình ở thanh ghi r4.
- Gọi User2System để chuyển tên chương trình từ vùng nhớ user space sang system space.
- Nếu bị lỗi không mở được file thì gán -1 vào thanh ghi r2.
- Nếu không lỗi thì gọi pTab->ExecUpdate(name) và lưu kết quả vào thanh ghi r2.

2. SC_Join

a. Chức năng:

- Thực hiện đợi tiến trình con dựa trên id của tiến trình con.

b. Cài đặt:

- Đọc id của tiến trình con từ thanh ghi r4.
- Gọi thực hiện pTab->JoinUpdate(id) và lưu kết quả thực hiện của hàm vào thanh ghi r2.

3. SC_Exit

a. Chức năng:

- Chương trình gọi sẽ thực hiện kết thúc tiến trình của nó hiện tại.

b. Cài đặt:

- Đọc exitStatus từ thanh ghi r4.
- Gọi thực hiện pTab->ExitUpdate(exitStatus) và lưu kết quả thực hiện của hàm vào thanh ghi r2.

4. SC_CreateSemaphore

a. Chức năng: Tạo Semaphore mới.

b. Cài đặt:

- Đọc địa chỉ “name” từ thanh ghi r4.
- Đọc giá trị “semval” từ thanh ghi r5.

- Gọi User2System để chuyển giả trị từ vùng nhớ của "name" hiện trong user space sang system space.
- Gọi semTab->Create(name, semval) để tạo semaphore.
- Lưu kết quả thực hiện vào thanh ghi r2.

5. SC_Up

a. Chức năng: giải phóng tiến trình đang chờ.

b. **Cài đặt:**

- Đọc địa chỉ "name" từ thanh ghi r4.
- Dùng User2Space chuyển giá trị vừa đọc từ vùng nhớ user space tới vùng nhớ system space.
- Kiểm tra Semaphore này có tồn tại trong bảng sTab hay không, nếu không thì báo lỗi.
- Nếu có gọi phương thức Signal của lớp Stable.
- Lưu kết quả thực hiện vào thanh ghi r2.

6. SC_Down

a. Chức năng: cho tiến trình vào trạng thái chờ.

b. Cài đặt:

- Đọc địa chỉ "name" từ thanh ghi r4.
- Lấy giá trị của "name" từ userspace vào system space bằng lệnh User2Space.
- Kiểm Semaphore "name" có trong bảng sTab chưa, nếu chưa thì báo lỗi.
- Gọi phương thức Wait() của lớp Stable.
- Lưu kết quả thực thi vào thanh ghi r2.

VI. Chương trình Shell

1. Mục đích:

- Dựa vào các lớp Ptable và PCB để chạy đa chương và đồng bộ hóa trên nachos.

2. Hoạt động:

- Chương trình hoạt động như chương trình Shell của hệ điều hành nhân UNIX. Sẽ nhận tên chương trình tại một thời điểm và thực thi.

3. Cài đặt:

- Chạy vòng lặp do … while(1) và sẽ dừng khi người dùng nhập lệnh "exit".
- Tạo promt để cho người dùng nhập tên chương trình.

- Kiểm tra chương trình có tồn tại hay không. Nếu không thì yêu cầu nhập lại.
- Thực thi chương trình.

VII. Chương trình ProduceConSume

1. Mục đích:

- Dựa vào các syscall SC_CreateSemaphore, SC_Up, SC_Down để thực hiện phối hợp hoạt động cho 2 chương trình produce và consume.

2. Hoạt động:

- Điều phốt hoạt động của 2 chương trình "produce" và "consume" theo các yêu cầu.
- Chương trình "produce" sẽ thực hiện xuất câu "Tạo ra 1 sản phẩm" 30 lần. Tương ứng với việc tạo ra 30 sản phẩm.
- Chương trình "consume" sẽ thực hiện xuất câu "dong goi 1 thung hang" 10 lần, tương ứng với việc đóng gói 10 thùng hàng.
- Yêu cầu :
    - Một thùng hàng phải có 3 sản phẩm.
    - Không thể đóng gói khi chưa có sản phẩm nào.
    - Một lần chỉ làm một sản phẩm hoặc chỉ đống 1 thùng hàng.