

HỆ ĐIỀU HÀNH

BÁO CÁO MULTIPROGRAMMING



Bộ môn Hệ điều hành
Khoa Công nghệ thông tin
Đại học Khoa học tự nhiên TP HCM

MỤC LỤC

1	Thông tin nhóm.....	2
2	Cài đặt.....	3
3	Các demo	7
4	Tài liệu tham khảo.....	8

1

Thông tin nhóm

MSSV	Họ Tên	Email	Điện thoại
1512003	Lê Tuấn Anh	ltanh035@gmail.com	0163 819 7063
1512004	Lê Việt Anh	levietanh.k15.it@gmail.com	0168 406 4269
1512029	Trần Quốc Bảo	tranquocbao3897@gmail.com	093 420 9840

2 Cài đặt

- Các lớp quan trọng về quản lý tiến trình cần phải tìm hiểu, và cài đặt: PTable và PCB là quan trọng nhất giúp quản lý các tiến trình thực thi. Cài đặt hai lớp này trong thư mục threads.
- 1. Lớp Thread: Ta thêm biến int processID để định danh các tiến trình đang chạy. Để ở public để cho dễ xử lý.
- 2. Lớp PCB: Lưu thông tin và phương thức để quản lý tiến trình.
 - Constructor: Khởi các giá trị ban đầu. Quan trọng là các biến Semaphore, với joinsem và exitsem thì khởi tạo giá trị ban đầu là 0, còn với mutex thì là 1. Vì joinsem và exitsem sẽ được sử dụng trong các thao tác chờ (JoinWait, ExitWait).

```
joinsem = new Semaphore("joinsem", 0);  
exitsem = new Semaphore("exitsem", 0);  
mutex = new Semaphore("mutex", 1);
```

Figure 1. Constructor PCB

- Hàm int Exec(char *filename, int pID): Hàm dùng để nạp tiến trình mới.
 - Tạo tiến trình mới.
 - Gán các tham số pid, processID của thread vừa tạo, parentID.
 - Gọi phương thức Fork của tiến trình vừa tạo.

```
mutex->P();  
thread = new Thread(filename);  
  
// Kiểm tra thread  
...  
  
// Set lại các tham số  
pid = pID;  
thread->processID = pID;  
parentID = currentThread->processID;  
  
thread->Fork(StartProcess_2, pID);  
mutex->V();
```

Figure 2. Exec - PCB

3. Lớp PTable: Quản lý các tiến trình đang chạy. Mảng lưu các PCB.

- Constructor: Quan trọng là phải dành phần tử đầu mảng cho tiến trình chính (main thread có processID = 0). Đồng thời khởi tạo các thuộc tính để sử dụng sau này.

```
// processID = 0 for main process
bm->Mark(0);
pcb[0] = new PCB(0);
pcb[0]->parentID = -1;
```

Figure 3. Constructor PTable

- Hàm int ExecUpdate(char* filename): đây là hàm dùng để nạp một tiến trình dựa trên đường dẫn tập tin thực thi. Làm theo hướng dẫn của giảng viên hướng dẫn thực hành. Có bốn bước quan trọng:
 - Xác thực đường dẫn tập tin thực thi và tập tin thực thi có hợp lệ hay không?
 - Kiểm tra xem nó có tự gọi chính nó hay không thông qua tên của tiến trình (cũng chính là đường dẫn tập tin thực thi).
 - Kiểm tra xem còn chỗ trống để thêm một tiến trình vào mảng để quản lý hay không?
 - Tạo mới một PCB (lớp PCB), gán parentID của lớp mới tạo bằng với processID của currentThread (chính là tiến trình đang gọi ExecUpdate). Sau đó gọi phương thức Exec của PCB.
- ❖ Lưu ý: Phải dùng bmsem để ngăn không cho tạo hai tiến trình cùng một lúc.
- Hàm int ExitUpdate(int ec): đây là hàm dùng để kết thúc tiến trình hiện tại (currentThread).
 - Nếu currentThread là main thread thì gọi interrupt->Halt(); sau đó return.
 - Nếu không phải, kiểm tra xem có tồn tại processID của currentThread có tồn tại trong mảng quản lý hay không? Nếu không thì return;
 - Nếu tồn tại thì SetExitCode cho PCB. Sau đó gọi phương thức JoinRelease để giải phóng tiến trình cha đang đợi nó kết thúc, xong gọi ExitWait() để xin tiến trình cha cho phép kết thúc.

- Sau đó giảm số tiến trình chờ của tiến trình cha. Rồi mới Remove tiến trình khỏi mảng quản lý.
- Hàm int JoinUpdate(int pID): để thực hiện đợi và block các tiến trình dựa trên định danh tiến trình pID.
 - Kiểm tra tính hợp lệ của pID. Không hợp lệ return.
 - Kiểm tra tiến trình gọi Join có phải là cha của tiến trình có processID là pID hay không. Không hợp lệ return.
 - Tăng số tiến trình chờ của tiến trình cha.
 - Gọi JoinWait để tiến trình cha chờ tiến trình con thực hiện xong.
 - Sau khi tiến trình con thực hiện xong, lấy exitcode của tiến trình con.
 - Gọi ExitRelease để cho phép tiến trình con kết thúc. Trả về exitcode.
- Sau khi có hai lớp PTable, PCB ta khai báo biến toàn cục PTable là pTab để sử dụng.
- Chỉnh sửa, thêm các hàm để xử lý việc cấp phát bộ nhớ cho các tiến trình:
- 1. Khai báo các biến sau: BitMap *gPhysPageBitMap – dùng để quản lý các trang vật lý. Semaphore *addrLock – dùng để ngăn không cho hai tiến trình cùng thực hiện thao tác nạp trang vật lý cùng lúc (vì có thể gây lỗi trong việc cấp phát trang).
- 2. Ở lớp AddrSpace: Ta thêm constructor AddrSpace(char *filename), đồng thời chỉnh sửa lại constructor cũ AddrSpace (OpenFile *executable) để phù hợp trong việc cấp phát đa chương.
- 3. Thêm hàm void StartProcess_2(int id) ở file proptest.cc để sử dụng kết hợp với Fork của thread.
 - Hàm này sẽ lấy id của tiến trình gọi Fork trong PTable.
 - Sau đó lấy đường dẫn tập tin thực thi thông qua phương thức GetName của PTable thông qua id.
 - Khởi tạo không gian địa chỉ vật lý thông qua OpenFile *executable sau khi gọi fileSystem->Open(filename);.
 - Gán không gian địa chỉ vật lý vừa tạo cho tiến trình đang chạy hiện tại.

4. Thêm các system call trong exception.cc:
- SC_Exec – dùng để gọi thực thi một chương trình mới trong một system thread mới.
 - SC_Exit – dùng để thực hiện thoát tiến trình đã join.
 - SC_Join – dùng để thực hiện đợi và block dựa trên SpaceID đã tạo ra lúc gọi SC_Exec.
- Viết chương trình để test hoạt động của phần đa chương vừa mới cài đặt xong. Như hướng dẫn của giảng viên hướng dẫn thực hành. Lưu ý: Trong scheduler.c phải thêm:

```
#include "syscall.h"

void main()
{
    int pingPID, pongPID;

    print("Ping-Pong test starting ...\n\n");
    pingPID = Exec("./test/ping");
    pongPID = Exec("./test/pong");

    Join(pingPID);
    Join(pongPID);
}
```

Figure 4. scheduler.c

- ✚ Như vậy, đã hoàn thành các nội dung đề án 02, bao gồm:
- Giải quyết các vấn đề quản lý, tạo và xóa các tiến trình.
 - Giải quyết các vấn đề cấp phát các frames bộ nhớ vật lý để cho nhiều chương trình có thể nạp lên bộ nhớ cùng lúc.
 - Biến hệ điều hành nachos từ đơn chương trở thành đa chương.

Các demo

[illegible]

Link youtube: <https://youtu.be/UMVMJBWoyJc>

4

Tài liệu tham khảo

Các file tài liệu do giảng viên hướng dẫn thực hành cung cấp. Bao gồm có:

- Huong Dan Cac Syscall Ve Da Chuong.pdf
- [5] Da Chuong Dong Bo Hoa.doc
- Seminar_HDH_Project 2.pptx
- Constructor_Cua_AddrSpace_2.pdf
- pcb.h
- ptable.h