

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

Đồ án

Đa chương

Chuyên ngành: Khoa học máy tính

GVHD: Phạm Tuấn Sơn

Mục Lục

I.	GIỚI THIỆU NHÓM:	3
II.	GIỚI THIỆU ĐỒ ÁN:	3
III.	Mức độ hoàn thành:	3
IV.	THIẾT KẾ VÀ CÀI ĐẶT:	3
1.	Cài đặt chi tiết:	3
a.	Cài thư mục và file đã chỉnh sửa:	3
b.	Cài syscall viết thêm:	4
c.	Nội dung cái file tạo thêm:	5
V.	CHẠY CHƯƠNG TRÌNH:	6
VI.	THAM KHẢO:	7
a.	Link video tham khảo:	7
b.	Link code tham khảo:	7
c.	Link report tham khảo:	8
d.	Link tài liệu tham khảo:	8

I. GIỚI THIỆU NHÓM:

STT	Họ và tên	MSSV
1	Phùng Tiến Hào	1612174
2	Phạm Phong Hào	1612176
3	Trần Thị Trúc Hân	1612170
4	Trần Thị Hồng Nhung	1612476

II. GIỚI THIỆU ĐỒ ÁN:

Tên đồ án: Đa chương

Giảng viên hướng dẫn: Phạm Tuấn Sơn

Thời gian thực hiện: 14 ngày

III. Mức độ hoàn thành:

STT	Tên	Mức độ (%)	Ghi chú
1	Cài đặt system call SpaceID Exec(char* name)		
a	Tạo ra một không gian địa chỉ mới	100%	
b	Load chương trình vào khoảng bộ nhớ mới được cấp phát	100%	
c	Sau đó tạo thread mới (bằng phương thức Thread::Fork()) để thực thi chương trình.	100%	
2	Cài đặt đa tiến trình.		
a	Giải quyết vấn đề cấp phát các frames bộ nhớ vật lý, sao cho nhiều chương trình có thể nạp lên bộ nhớ cùng một lúc.	100%	
b	Phải xử lý giải phóng bộ nhớ khi user program kết thúc.	100%	
c	Phản quan trọng là thay đổi đoạn lệnh nạp user program lên bộ nhớ.	100%	

IV. THIẾT KẾ VÀ CÀI ĐẶT:

1. Cài đặt chi tiết:

a. Cái thư mục và file đã chỉnh sửa:

STT	Tên thư mục	Nội dung đã thêm vào và chỉnh sửa
-----	-------------	-----------------------------------

1	thread.h	Bổ sung thêm một thuộc tính mới là: ProcessID dùng để phân biệt các tiến trình với nhau
2	thread.cc	Bổ sung khởi tạo giá trị cho các thuộc tính mới thêm vào
3	system.h	Bổ sung các biến toàn cục sau: <ul style="list-style-type: none"> extern Semaphore *addrLock; // semaphore trong synch.h, dùng để extern BitMap *gPhysPageBitMap; // quan ly cac frame vat ly tren ram extern BitMap *pageTable; // Quan ly cac trang extern int *parentIdTable; // Quan ly cac tien trinh cha extern char** fileNameTable; // Quan ly ten cac tien trinh Lưu ý: biến currentThread là con trỏ lưu giữ tiến trình hiện tại.
4	system.cc	Cấp phát bộ nhớ cho các biến toàn cục vừa khai báo ở trên
5	synch.h / synch.cc	Dùng phương thức P() để chuyển tiến trình hiện tại sang trạng thái chờ và đợi tiến trình con thực hiện
6	addrspace.h	Bổ sung thêm hàm khởi tạo AddrSpace(char* filename); // Bổ sung phương thức khởi tạo với tham số truyền vào: filename
7	addspace.cc	Chỉnh sửa hàm khởi tạo AddrSpace(OpenFile *executable), bổ sung thêm hàm khởi tạo AddrSpace(char* filename) và chỉnh sửa hàm ~AddrSpace()
8	bitmap.h	Sử dụng bit map để quản lý các tiến trình.
9	bitmap.cc	
10	protest.cc	Thêm phương thức StartProcess_2 và chỉnh sửa phương thức StartProcess Ý nghĩa: Khi 1 tiến trình chạy thì nó sẽ thực hiện kiểm tra và cấp phát bộ nhớ
11	syscall.h	Định nghĩa các syscall và hàm cho Exec, Wait
12	exception.cc	Cài đặt các syscall SC_Exec, SC_Wait

b. Cái syscall viết thêm:

STT	Tên syscall	Nội dung
1	SC_Exec	// a. Tạo ra một không gian địa chỉ mới Thread *mythread; int pid; // Kiểm tra thread đã khởi tạo thành công chưa, nếu chưa thì báo lỗi là không đủ bộ nhớ và return -1 mythread = new Thread(name); // (./threads/thread.h) // ... // b. Load chương trình vào khoảng bộ nhớ mới được cấp phát // ...

		// c. Sau đó tạo thread mới (bằng phương thức Thread::Fork()) để thực thi chương trình. // Gọi thực thi Fork(StartProcess, pid) => Ta cast thread thành kiểu int, sau đó khi xử lý hàm StartProcess ta cast Thread về đúng kiểu của nó. mythread->Fork(StartProcess_2, pid); // Nhường CPU cho mythread currentThread->Yield(); // ...
2	SC_Wait	// ... addrLock->P(); // Tiến trình chuyển sang trạng thái block và ngừng lại, đợi tiến trình con thực hiện

c. Nội dung cái file tạo thêm:

File schedules.c trong thư mục test

```
#include "syscall.h"
void main()
{
    int pingPID, pongPID;
    PrintString("Ping-Pong test starting...\n\n");
    pingPID = Exec("./test/ping");
    pongPID = Exec("./test/pong");
    Wait(pingPID);
    Wait(pongPID);
    PrintString("\n\n");
}
```

File ping.c trong thư mục test.

```
#include "syscall.h"
int main()
{
    int i;
    for(i = 0; i < 1000; i++)
    {
        PrintChar('A');
    }
}
```

File pong.c trong thư mục test.

```
#include "syscall.h"
```

```
int main()
{
    int i;
    for(i = 0; i < 1000; i++)
    {
        PrintChar('B');
    }
}
```

V. CHẠY CHƯƠNG TRÌNH:

```
[sv@localhost code]$ ./userprog/nachos -rs 1023 -x ./test/scheduler
```

```
Size: 1536 | numPages: 3 | PageSize: 512 | Numclear: 256
```

```
Physic Pages 0
```

```
Physic Pages 1
```

```
Physic Pages 2
```

```
Ping-Pong test starting...
```

```
Size: 1536 | numPages: 3 | PageSize: 512 | Numclear: 253
```

```
Physic Pages 3
```

```
Physic Pages 4
```

```
Physic Pages 5
```

```
A
```

```
Size: 1536 | numPages: 3 | PageSize: 512 | Numclear: 250
```

```
Physic Pages 6
```

```
Physic Pages 7
```

```
Physic Pages 8
```

```

Size: 1536 | numPages: 3 | PageSize: 512 | Numclear: 250

Physic Pages 6
Physic Pages 7
Physic Pages 8

AAAAAAAAABBBBBABBBBBBABBBBBBAAAAAAAAABBAABBBBAAABBBABBAABBBBBBA
AABBBBBBAAABABBABBBAAABBBBBBABBBBBBAAABBAABAAAABBBABAAAABBBABAAAAB
ABAABBBBBBAAAAAAAAAABABAABBBBBBAAAAABBBABBBBBBBBAAABBAABAAAAAABAAAA
BBBBBBBAABBBBBBAABBBAAAAABBAAAAAAAAAABBBBBBBBBBBBAAAAABBBBBBBBBBBBA
AABBBBBAAAAAAAABBBBBBBBBAABAABBAAAAAAAAAAAAAABBBBABBBBBBBBBBAAAAABBB
AAAAAAAAAAAAABBBBBBBBBBBBAAAAAABBAAAAAABBAAAAAABBBAAAAAABAAAAAABBB
BAAAABBBBBBAAAAABBAAAAAAABAABBAAAAAAABBBBAAAABBBBAAAABBBBBBBBAB
BBBAAAAAABBBBBBBBBBBBBBBBBBBBBAABBBBBBBBAAAABBBBBBBBBBAAAAAABBB
BBBBBBBBBBBAABAABBAABBAABBAABBBBAAAABBBBAAAAAAAAAAAAAAAAABBBAAAAAA
BBABBBBBAAAAABAABAABBBBAAABBBBBBBBBBBBAAAAABAAAABAAAABBBAAAAABBB
BBBBBAAAAAAAAAAAAABBAAAAABBBBBBAAAAAAAAAAAAAAAAAAAAABBBAAAABBBBBB
AABBBBBBBBBBBBAAAAAAAABBAABBBBAAABBBBAAABBBBAAAAAAAABAAAABBBBBBBBBB

Shutdown, initiated by user program.Machine halting!

Ticks: total 282383, idle 184935, system 66670, user 30778
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 1947
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
[sv@localhost code]$ █

```

VI. THAM KHẢO:

- Link video tham khảo:
<https://www.youtube.com/channel/UCppAv0JKFKROP2goBfBM5og>
- Link code tham khảo:
<https://github.com/nguyenthanhchungfit/Nachos-Programing-HCMUS>

- c. Link report tham khảo:
https://drive.google.com/drive/folders/1UtGBISw5L_071_HGMYDI2Qs6n5q5-tFO
- d. Link tài liệu tham khảo:
NachOS Programming Project -Vania Marangozova-Marti - 2009-2010