**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](#_Toc531810228)

[**II.** **GIỚI THIỆU ĐỒ ÁN:** 3](#_Toc531810229)

[**III.** **Mức độ hoàn thành:** 3](#_Toc531810230)

[**IV.** **THIẾT KẾ VÀ CÀI ĐẶT:** 3](#_Toc531810231)

[**1.** **Cài đặt chi tiết:** 3](#_Toc531810232)

[**a.** **Cái thư mục và file đã chỉnh sửa:** 3](#_Toc531810233)

[**b.** **Cái syscall viết thêm:** 4](#_Toc531810234)

[**c.** **Nội dung cái file tạo thêm:** 5](#_Toc531810235)

[**V.** **CHẠY CHƯƠNG TRÌNH:** 6](#_Toc531810236)

[**VI.** **THAM KHẢO:** 7](#_Toc531810237)

[a. Link video tham khảo: 7](#_Toc531810238)

[b. Link code tham khảo: 7](#_Toc531810239)

[c. Link report tham khảo: 8](#_Toc531810240)

[d. Link tài liệu tham khảo: 8](#_Toc531810241)

# **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 |

# **GIỚI THIỆU ĐỒ ÁN:**

*Tên đồ án:* **Đa chương**

*Giảng viên hướng dẫn:*Phạm Tuấn Sơn

*Thời gian thực thiện:*14 ngày

1. **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 | Phần quan trọng là thay đổi đoạn lệnh nạp user program lên bộ nhớ. | 100% |  |

# **THIẾT KẾ VÀ CÀI ĐẶT:**

## **Cài đặt chi tiết:**

### **Cái thư mục và file đã chỉnh sửa:**

|  |  |  |
| --- | --- | --- |
| STT | Tên thự 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:   * 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 | Dung phuong thuc 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  bitmap.cc | Sử dụng bit map để quản lý các tiến trình. |
| 9 |
| 10 | protest.cc | Thêm phương thức StartProcess\_2 va 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 |

### **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 va 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 ký hàm StartProcess ta cast Thread về đúng kiểu của nó.  mythread->Fork(StartProcess\_2, pid);  // Nhuong 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 |

### **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");  } |

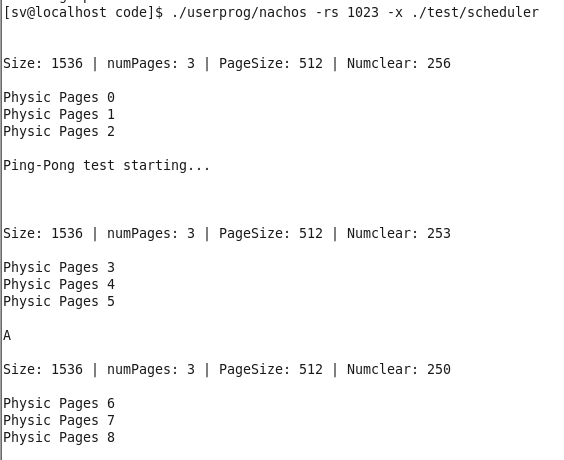
File ping.c trong thư mục test.

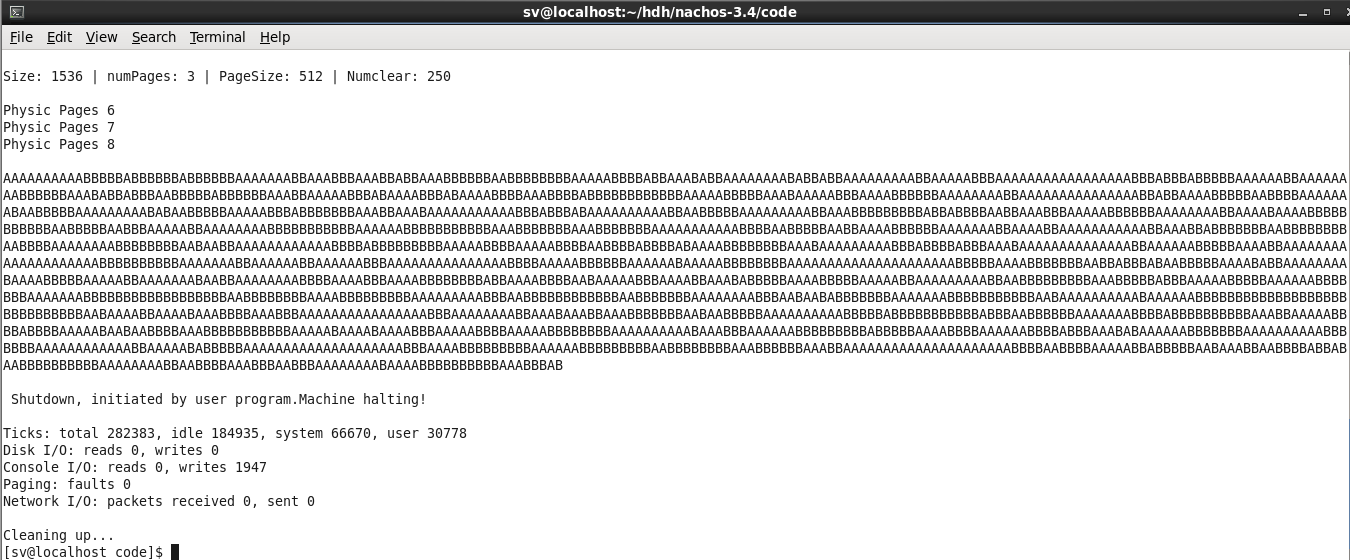
|  |
| --- |
| #include "syscall.h"  int main()  {  int i;  for(i = 0; i < 1000; i++)  {  PrintChar('A');  }  } |

File pong.c trong thư mục test.

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

# **CHẠY CHƯƠNG TRÌNH:**





# **THAM KHẢO:**

### Link video tham khảo:

<https://www.youtube.com/channel/UCppAv0JKFKROP2goBfBM5og>

### Link code tham khảo:

<https://github.com/nguyenthanhchungfit/Nachos-Programing-HCMUS>

### Link report tham khảo:

<https://drive.google.com/drive/folders/1UtGBlSw5L_071_HGMYDI2Qs6n5q5-tFO>

### Link tài liệu tham khảo:

NachOS Programming Project -Vania Marangozova-Marti - 2009-2010