ĐẠI HỌC QUỐC GIA TPHCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo bài tập

Đề tài: xv6

Môn học: Operating System

Sinh viên thực hiện:

Phạm Quốc Nam Anh	23120111
Nguyễn Xuân Duy	23120121
Nguyễn Minh Hiếu	23120124
Huỳnh Thái Toàn	23120175

Giáo viên hướng dẫn:

Lê Viết Long

Ngày 10 tháng 12 năm 2025

Mục lục

1 Tăng tốc syscall

1.1 Mô tả bài toán

Tối ưu hóa lệnh gọi hệ thống `getpid()` bằng cách chia sẻ một vùng nhớ chỉ đọc giữa kernel và user. Thay vì trap vào kernel mỗi lần gọi getpid, tiến trình user có thể đọc trực tiếp PID từ một trang bộ nhớ đặc biệt đã được ánh xạ sẵn. Trang này chứa một struct `usyscall` với trường `pid` lưu PID của tiến trình hiện tại.
Ngoài ra, viết chương trình kiểm tra `pidget()` để so sánh giá trị trả về từ getpid() (system call thông thường) và ugetpid() (đọc trực tiếp từ vùng chia sẻ). Bài tập đạt yêu cầu nếu hai giá trị này trùng nhau.

1.2 Phương pháp thực hiện

1. Trong `proc_pagetable()`, khởi tạo page table mới và ánh xạ các vùng bắt buộc (trampoline, trapframe).
2. Trong `allocproc()`, cấp phát và khởi tạo tiến trình, đồng thời xử lý cấp phát và ánh xạ trang `USYSCALL`.
3. Trong `freeproc()`, giải phóng tài nguyên tiến trình, bao gồm cả trang `USYSCALL` nếu tồn tại.

1.2.1 Khởi tạo page table

Trong hàm `proc_pagetable()`:

- Tạo một page table mới bằng `uvmcreate()`.
- Ánh xạ `trampoline` và `trapframe` vào không gian địa chỉ user.
- Đảm bảo các ánh xạ này có quyền truy cập phù hợp.

1.2.2 Cấp phát và khởi tạo tiến trình

Trong hàm `allocproc()`:

- Tìm một tiến trình rảnh trong mảng `proc[]` và khởi tạo các trường cơ bản (PID, trạng thái, trapframe).

- Cấp phát một trang cho `usyscall` bằng `kalloc()`.
- Khởi tạo trường `pid` trong `struct usyscall`.
- Ánh xạ địa chỉ ảo `USYSCALL` tới trang vật lý chứa `struct usyscall` bằng `mappages()` với quyền chỉ đọc cho user.

1.2.3 Giải phóng tiến trình

Trong hàm `freeproc()`:

- Giải phóng vùng nhớ được cấp cho `usyscall` nếu tồn tại.
- Đặt lại trạng thái tiến trình về `UNUSED`.

Trong hàm `proc_freepagetable()`:

- Bỏ ánh xạ vùng `USYSCALL` khỏi page table bằng `uvmunmap()`.
- Giải phóng toàn bộ page table của tiến trình.