Operating systems

**Project2 report**

**رضا چهرقانی 810101401**

**امیر نداف فهمیده 810101540**

**مصطفی کرمانی‌نیا 810101575**

**مخزن گيتهاب اين پروژه‌:**

**https://github.com/mostafa-kermaninia/OS\_LAB\_P2**

horizontal line

* **پرسش 1) کتابخانه‌های سطح کاربر در XV6، برای ایجاد ارتباط میان برنامه‌های کاربر و کرنل به کار می‌روند. این کتابخانه‌ها شامل توابعی هستند که از فراخوانی‌های سیستمی استفاده می‌کنند تا دسترسی به منابع سخت‌افزاری و نرم‌افزاری سیستم‌عامل ممکن شود. با تحلیل فایل‌های موجود در متغیر ULIB در XV6، توضیح دهید که چگونه این کتابخانه‌ها از فراخوانی‌های سیستمی بهره می‌برند؟ همچنین، دلایل استفاده از این فراخوانی‌ها و تأثیر آنها بر عملکرد و قابلیت حمل برنامه‌ها را شرح دهید.**
* **پرسش 2) فراخوانی‌های سیستمی تنها روش برای تعامل برنامه‌های کاربر با کرنل نیستند. چه روش‌های دیگری در لینوکس وجود دارند که برنامه‌های سطح کاربر می‌توانند از طریق آنها به کرنل دسترسی داشته باشند؟ هر یک از این روش‌ها را به اختصار توضیح دهید.**

می‌توانیم وقایع رخ داده در سیستم را به صورت زیر دسته‌بندی کنیم:

۱- Exception  
۲- Interrupt

* S.W interrupt
* H.W interrupt

به طور کل دسترسی به هسته با یک interrupt (وقفه) رخ می‌دهد.

وقفه سخت‌افزاری: این وقفه توسط دستگاه‌های سخت‌افزاری خارجی به صورت آسینکرون تولید می‌شود که به طور مثال می‌تواند برای ورودی کاربر، تکمیل عملیات I/O و... باشد. در حالتی که چنین وقفه‌ای رخ می‌دهد CPU فعالیت فعلی خود را ذخیره می‌کند و کنترل را به interrupt service routine منتقل می‌کند. بعد از انجام کارهای مربوطه برای رفع interrupt کنترل به CPU باز می‌گردد.

وقفه نرم‌افزاری (trap): این وقفه توسط برنامه به صورت سنکرون ایجاد می‌شود. این وقفه‌ها معمولاً به دلیل درخواست انجام یک کار توسط سیستم عامل ایجاد می‌شود که به طور مثال می‌توان به درخواست یک حافظه اشتراکی، خاتمه دادن به یک برنامه، بازکردن یک فایل و... باشد.

از انواع trap ها می‌توان به:  
1- Signal  
سیگنال های مختلف در لینوکس وجود دارد نظیر SIGKILL  
 System call -2  
که به آن اشاره شده است.

در لینوکس Pseudo-file-systems نیز وجود دارد.

* **پرسش 3) آیا باقی تله‌ها را نمی‌توان در سطح DPL USER فعال نمود؟ چرا؟**

خیر چنین کاری امکان ندارد. سطح ذکر شده سطح دسترسی کاربر است و در این سطح نباید اجازه دسترسی به هسته سیستم عامل (کرنل) و اجرای تله‌ها را داشت. درصورت انجام این عمل protection exception فعال می‌شود.

لازم به ذکر است که درصورتی که این اجازه داده می‌شد به kernel protection ایراد وارد می‌شد.

از دلایلی که این موضوع در انحصار سطح kernel است می‌توان به:

۱- یک برنامه قصد سوءاستفاده از هسته را دارد و با این روش کنترل کل هسته سیستم عامل را در اختیار می‌گیرد.

۲- یک برنامه مخرب در چنین حالتی می‌تواند با آسیب به هسته تمام سطوح H.W & S.W آسیب وارد کند.

۳- اگر برنامه کاربر دچار ایراد باشد در حالتی که در سطح kernel اجرایی شود این مشکل می‌تواند به تمام سطح هسته گسترش یابد.

* **پرسش 4) در صورت تغییر سطح دسترسی، ss و esp روی پشته Push می‌شود. در غیر این‌صورت Push نمی‌شود. چرا؟**
* **پرسش 5) در مورد توابع دسترسی به پارامترهای فراخوانی سیستمی به طور مختصر توضیح دهید. چرا در argptr() بازه آدرس‌ها بررسی می‌گردد؟ تجاوز از بازه معتبر، چه مشکل امنیتی ایجاد می‌کند؟ در صورت عدم بررسی باز در این تابع، مثالی بزنید که در آن، فراخوانی سیستمی read\_sys() اجرای سیستم را با مشکل روبرو سازد.**