

سیستم‌های عامل

اضافه کردن فراخوان سیستمی به Xv6

تعریف پروژه

به سیستم عامل Xv6 یک فراخوانی سیستمی^۱ جدید اضافه کنید. نام فراخوانی سیستمی را `proc_dump` قرار دهید. در نتیجه اجرای `proc_dump` باید اطلاعات پروزه‌های در حال اجرا (در حالت `RUNNING` و یا `RUNNABLE`) را به صورت یک آرایه در اختیار کاربر قرار دهد. آرایه مورد نظر از جنس `struct proc_info` است و در ادامه تعریف شده است. آرایه باید بر حسب مقدار `memsize` مرتب شده باشد (الگوریتم مرتب سازی مهم نیست).

```
1 struct proc_info {  
2     int pid;  
3     int memsize; // in bytes  
4 }
```

فراخوانی سیستمی که تعریف می‌کنید می‌تواند آرایه‌ای از جنس `struct proc_info` را به همراه اندازه آن از کاربر دریافت کند و محتوای آن را پرکند. با این روش نیازی به درخواست حافظه در خود فراخوانی سیستمی وجود ندارد.

بعد از تعریف فراخوانی سیستمی یک برنامه کاربردی بنویسید که این فراخوان سیستمی را تست کند. در این برنامه تست، با استفاده از `malloc` و `fork` تعدادی پروزه متفاوت ایجاد کنید و مقدار متفاوت حافظه به هر یک اختصاص دهید. با این کار در هنگام فراخوانی باید تعدادی پروزه در حال اجرا با مقدار حافظه متفاوت در نتیجه مشاهده شود.

نکات:

۱. یک فراخوان سیستمی جدید تعریف کنید. کنید که اطلاعات پروزه‌های در حال اجرا را برگرداند.

۲. نام فراخوان سیستمی را `proc_dump` قرار دهید.

۳. خروجی این فراخوان سیستمی یک آرایه مرتب شده بر حسب `memsize` است.

۴. فایل‌هایی که کامپایل می‌شود در `Makefile` این پروژه تعریف شده است. برنامه تست خود را باید به بخش `UPROGS` اضافه کنید. اگر فایل دیگری نیز به کرنل اضافه کرده‌اید باید در مکان مناسب در `Makefile` تعریف کنید.

^۱System call

۵. توجه کنید malloc را در لایه کرنل نمی‌توانید استفاده کنید چرا که خود malloc دارد از فراخوانی سیستمی استفاده می‌کند و برای برنامه‌های کاربردی پیاده‌سازی شده است.

ارائه پروژه

برای پروژه خود یک گزارش آماده کنید و چگونگی پیاده‌سازی خود و مراحل اجرای فراخوانی سیستمی را در آن شرح دهید. سپس تمام فایل‌هایی که تغییر داده‌اید و یا خودتان ایجاد کرده‌اید را همراه با گزارشتان به صورت zip و با نام os_proj1_stdnum بر روی سایت کوئرا آپلود نمایید.

آماده‌سازی محیط کار بار Xv6

در ادامه راهنمایی برای آماده‌سازی محیط کار خود با Xv6 ارائه می‌شود. این مراحل با سیستم عامل Debian10.9 تست شده است و برای سیستم‌های دیگر مراحل مشابه باید طی شود. در صورت وجود مشکل در آماده‌سازی محیط کار با Xv6 توصیه می‌شود با ایجاد یک ماشین مجازی محیط کار خود را آماده کنید.

ابتدا مطمئن شوید که نیازمندی‌های لازم برای کامپایل کردن و اجرا بر روی سیستم شما نصب است. برای اجرای Xv6 از یک emulator به اسم qemu استفاده می‌شود.

```
sudo apt-get update && sudo apt-get install --yes build-essential git  
qemu-system-x86
```

سپس سورس سیستم عامل Xv6 را از این لینک^۲ دریافت کنید.

```
git clone https://github.com/mit-pdos/xv6-public
```

با استفاده از دستور make می‌توانید کرنل را کامپایل کنید.

```
make qemu
```

^۲<https://github.com/mit-pdos/xv6-public>