



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

پروژه - فاز سوم

دکتر جوادی

سیستم‌های عامل

مقدمه:

در فاز آخر پروژه درس سیستم‌عامل، هدف اصلی شما تغییر و بهبود سیستم زمان‌بندی (Scheduler) در کرنل آموزشی xv6 است. برای دستیابی به این هدف، ابتدا باید ساختار فعلی Scheduler در xv6 را به خوبی درک کرده و سپس با مدیریت cpuusage هر پردازش آن را مدیریت کنید.

آشنایی با مفاهیم اولیه:

۱. Scheduler

- کرنل xv6 به طور پیش‌فرض از الگوریتم **Round-Robin** برای زمان‌بندی استفاده می‌کند. این الگوریتم به صورت چرخشی به هر پردازش زمانی برابر برای اجرا اختصاص می‌دهد.
- فانکشن Scheduler در فایل `kernel/proc.c` تعریف شده و مسئول جابجایی بین پردازش‌ها است.

چه قابلیت هایی میخوایم پیاده سازی کنیم؟

در این فاز میخواهیم ابتدا بتوانیم ذخیره کنیم که هر پردازش چقدر از cpu استفاده کرده است. (چند tick را consume کرده است) و در نهایت با توجه به این میزان یک scheduler بهینه تر بنویسیم به طوری که هر پردازش ای که کمتر از cpu استفاده کرده ابتدا اولویت داده شود.

همچنین سه سیستم کال زیر را نیز به پروژه اضافه میکنیم:

❖ **cpu_usage**

یک سیستم کال که میزان استفاده از cpu پردازش فعلی را برمیگرداند.

❖ **top**

لیستی از پردازش های فعال به همراه میزان استفاده آنها (سورت شده بر اساس میزان استفاده) و استیت آنها نشان میدهد.

❖ **set_cpu_quota**

یک پردازش میتواند با این سیستم کال میزان استفاده از cpu پردازش خود و یا فرزندان و نوادگان خود را محدود کند.

به چه استراکت های جدیدی نیاز داریم؟

برای پیاده سازی لازمه که استراکت زیر رو به proc.h اضافه کنیم:

```
struct cpu_usage {  
    uint sum_of_ticks;  
    uint start_tick;  
    uint quota;  
};
```

زمانی که یک پردازش شروع به اجرا می شود باید start_tick رو ست کنیم و زمانی که وارد ترپ می شود (جایی که از پردازش خارج شده است) تفاضل tick الان cpu و start_tick را باید به sum اضافه کنیم. در نهایت این میزان sum همان میزان استفاده از cpu پردازش را نشان میدهد.

در نهایت یک instance از این استراکت در استراکت proc نگه داری کنید.

سیستم کال Top:

در این بخش قرار است که یک سیستم کال جدید به اسم top به سیستم عامل آموزشی XV6 اضافه کنیم. این سیستم کال لیستی از اطلاعات پردازش های در حال اجرا را بر میگرداند.

```
int top(struct top*);
```

به طور کلی یک همچین سیستم کالی خواهیم داشت که ورودی پوینتر به یک struct میگیرد تا آن را پر کند. اگر با موفقیت انجام شد خروجی ۰ میدهد. برای struct top همچین چیزی پیشنهاد می شود: (اسم و اطلاعات آن را میتوانید با توجه به نیاز های خود تغییر دهید).

```
struct proc_info {  
    char name[16];  
    int pid;  
    int ppid;  
    enum procstate state;  
    struct cpu_usage usage;  
};
```

```
struct top {  
    int count;  
    struct proc_info processes[NPROC];  
};
```

خروجی مورد انتظار چیزی مشابه زیر خواهد بود:

```
$ top
number of process: 16
PID      PPID     STATE   NAME    START   USAGE
6         5        sleep   top     100     92
7         6        sleep   top     230     51
8         6        sleep   top     383     48
9         6        sleep   top     490     39
```

دقت کنید که خروجی بر اساس USAGE سورت شده باشد.

سیستم کال `set_cpu_quota`:

با استفاده از این سیستم کال می توانیم میزان استفاده از cpu خود و یا فرزندان و نوادگان خود را محدود کنیم.

```
int set_cpu_quota(int pid, int quota);
```

چگونه scheduler را تغییر دهیم؟

به طور پیشفرض xv6 به صورت حلقه ای روی پردازش ها جلو میرود و به هر پردازش آزاد assign می شود. شما اکنون با دانش آن که cpu_usage هر پردازش چقدر است باید اولویت را به پردازش ای بدهید که میدانید cpu_usage کمتری دارد.

اگر میزان استفاده پردازش از quota مشخص شده بیشتر شده باشد باید به یک صف جدید (صف پردازش های کم اولویت اضافه شود) و در نهایت اگر هیچ پردازش ای برای اجرا وجود نداشت دوباره الگوریتم اسکرولر روی صف پردازش هایی که اولویت کمی داشتند اجرا شود و این بار محدودیت quota در نظر گرفته نشود.

تست:

برای تست کردن برنامه خود میتوانید از fork استفاده کنید و چند پردازش ایجاد کنید و سپس با sleep کردن پردازش cpu_usage آن را کم کنید. همچنین در برنامه تست شما باید سیستم کال top نیز بررسی شود و صحت عملکرد آن مشخص شود.

توضیحات پایانی:

از شما درخواست داریم که یک **private repository** در گیت هاب درست کنید و تغییرات کد خود را مرحله به مرحله **Commit** کنید و در صورت تمایل می‌توانید هر یک از تدریس یاران را به پروژه خود اضافه کنید. دقت کنید که شما نبایستی برنامه‌های خود را با دیگر دانشجویان به اشتراک بگذارید.

توضیحات

- یک نفر از اعضای گروه کفایت که پروژه را آپلود کند.
- پروژه شما تحویل آنلاین خواهد داشت بنابراین از استفاده از کدهای دیگران یا کدهای موجود در وب که قادر به توضیح دادن عملکرد آنها نیستید، بپرهیزید.
- ابهامات خود را در گروه درس در تلگرام مطرح کنید و ما در سریع‌ترین زمان ممکن به آنها پاسخ خواهیم داد.

آنچه که باید ارسال کنید:

- یک فایل زیپ با نام **OS_P3_Sid1_Sid2.zip** که شامل دو مورد زیر است:
 1. گزارش خیلی مختصر از آنچه که انجام داده‌اید.
 2. پوشه ای که در آن کدهای شما وجود دارد. دقت کنید که تنها و تنها فایل‌هایی را که تغییر داده‌اید یا اضافه کرده‌اید را برای ما بفرستید.

موفق باشید

تیم تدریس یاری درس سیستم های عامل