

به نام ایزد یکتا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

فاز اول پروژه درس سیستم عامل



دانشکده مهندسی کامپیوتر

استاد: جوادی

تهیه کننده: بردیا اردکانیان

۹۸۳۱۰۷۲

بخش اول: فراخوانی سیستمی ticks

همانطور که می‌دانید هر سیستم عاملی برای جلو رفتن پردازش‌هایش به کلاک نیاز دارد. در واقع مانند ثانیه شمار عمل می‌کند (مفهوم انتزاعی دارد و به معنی دقیق ثانیه نمی‌باشد). متغیر ticks که به مفهوم تیک تیک ساعت نیز می‌باشد کلاک‌هایی که تا کنون گذشته است را ذخیره می‌کند. این متغیر در فایل trap.c با آمدن وقفه timer آپدیت می‌شود و یک واحد افزایش می‌یابد.

به جهت برگرداندن این مقدار نیاز است فراخوانی سیستمی اضافه کنیم که این مقدار را بخواند و برگرداند. تنها نیاز است در تابع این فراخوانی سیستمی return ticks; را بنویسیم تا کار را انجام دهد. به همین جهت فراخوانی سیستمی به نام recticks (receive ticks) را اضافه کرده‌ام و در آن مقدار ticks را برگردانده‌ام.

همچنین برای تست این فراخوانی سیستمی یک فایل ticktest.c ساختم که این فراخوانی سیستمی را صدا می‌زند و نتیجه را در ترمینال به نمایش می‌گذارد.

```
// recives the number of ticks
int
sys_recticks(void)
{
    return recticks();
}

// recieve number of system ticks
int
recticks(void)
{
    return ticks;
}
```

بخش دوم: فراخوانی سیستمی procinfo

در این فراخوانی سیستمی قرار است که در بین تمامی پردازش‌های سیستم پیمایش کند و pid و زمان ایجاد پردازش‌هایی که در وضعیت RUNNING هستند را چاپ می‌کند. برای این کار ابتدا می‌بایست متغیر تایم (ctime) را در ساختار پردازش‌ها اضافه می‌کنیم و در لحظه ساخته شدن پردازش مقدار ticks که در بخش پیش در مورد آن صحبت کردیم را در آن ذخیره می‌کنیم. بعد با استفاده از for بر روی همه پردازش‌ها پیمایش می‌کنیم و اطلاعات pid و ctime پردازش‌هایی که در وضعیت RUNNING هستند را چاپ می‌کنیم.

همچنین یک فایل proctest.c ساخته که صحت فراخوانی سیستمی نوشته شده را بسنجد.

```
// Iterate through process and display RUNNING process pid, time
int
procinfo(void)
{
    struct proc *p;

    acquire(&ptable.lock);
    for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
        if(p->state == RUNNING) {
            cprintf("%d    %d\n", p->pid, p->ctime);
        }
    }

    release(&ptable.lock);
    return 0;
}
```