

بسمه تعالی



گزارش تمرین عملی چهارم - سوال اول - درس سیستم‌های عامل دکتر اسدی - نیمسال اول ۱۴۰۵-۱۴۰۴
نویسنده‌گان:

۱. سیداحمد موسوی‌اول - ۰۲۱۰۶۶۴۸ - ۱۴۰۲

۲. عرفان تیموری - ۰۲۱۰۵۸۱۳ - ۱۴۰۲

در این تمرین باید یک عملکردی به عنوان context switch برای تردها درست کنیم و تنها فایلی که نیاز به تغییر آن داریم فایل `uthread.c` است.

در این تمرین، مدیریت نخ‌های کاربر را در سیستم عامل XV6 پیاده‌سازی کردیم. هدف اصلی این بود که بتوانیم چندین نخ کاربر را ایجاد کنیم و بین آن‌ها تعویض زمینه context switch انجام دهیم.

ساختارهای داده:

```
1. struct context {
2.     uint64 ra; // آدرس بازگشت
3.     uint64 sp; // اشاره‌گر پشته
4.     // ثبات‌های callee-saved
5.     uint64 s0, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11;
6. };
7.
8. struct thread {
9.     char stack[STACK_SIZE]; // پشته نخ
10.    int state; // وضعیت نخ (FREE, RUNNING, RUNNABLE)
11.    struct context context; // زمینه نخ برای تعویض
12.};
```

تابع اصلی:

۱. `Thread_init`: مقداردهی اولیه، مدیریت نخ‌ها و همچنین تنظیم نخ اصلی به عنوان نخ در حال اجرا
۲. `thread_create(void (*func)())`: یافتن یک نخ آزاد در آرایه، تنظیم وضعیت به `RUNNABLE`، تنظیم پشته و زمینه برای نخ جدید، تنظیم اشاره‌گر پشته (`SP`) و آدرس بازگشت (`RA`)
۳. `thread_schedule()`: زمان‌بندی راند-روبن بین نخ‌ها، یافتن نخ بعدی قابل اجرا، انجام تعویض زمینه `thread_switch` با `thread_switch`
۴. `thread_yield()`: واگذاری CPU به نخ دیگر، تغییر وضعیت به `RUNNABLE` و فراخوانی زمان‌بند

برای تست و خروجی هم موارد زیر را انجام دادیم:

سه نخ thread-a، thread-b و thread-c ایجاد شدند که هر کدام:

- پیام شروع را چاپ می‌کنند

- منتظر شروع سایر نخها می‌مانند

- 100 بار حلقه می‌زنند و در هر بار شماره خود را چاپ می‌کنند

- در پایان پیام خروج را چاپ می‌کنند

```
thread_a_90
thread_b_90
thread_c_91
thread_a_91
thread_b_91
thread_c_92
thread_a_92
thread_b_92
thread_c_93
thread_a_93
thread_b_93
thread_c_94
thread_a_94
thread_b_94
thread_c_95
thread_a_95
thread_b_95
thread_c_96
thread_a_96
thread_b_96
thread_c_97
thread_a_97
thread_b_97
thread_c_98
thread_a_98
thread_b_98
thread_c_99
thread_a_99
thread_b_99
thread_c: exit after 100
thread_a: exit after 100
thread_b: exit after 100
thread_schedule: no runnable threads
$ █
```

کد uthread.c هم در کنار همین فایل قرار دارد.