



فهرست

۱	دستورکار شماره دو:
۱	اهداف آموزشی:
۲	توضیح پروژه:
۲	۱-۳ پیش‌نیازها و ابزار موردنیاز:
۲	۲-۳ ساختار کلی پروژه Pintos:
۲	۳-۳ مراحل انجام آزمایش:
۲	۱-۳-۳ فاز اول - آماده‌سازی و شناخت ساختار
۲	۲-۳-۳ فاز دوم - طراحی:
۲	۳-۳-۳ فاز سوم - ساعت زنگ‌دار (Alarm Clock):
۳	۴-۳-۳ فاز چهارم - زمان‌بند اولویت‌محور (Priority Scheduler):
۳	۵-۳-۳ فاز پنجم - زمان‌بند پیشرفته (Advanced Scheduler - MLFQS):
۳	۶-۳-۳ فاز ششم - تست و ارزیابی:
۳	۷-۳-۳ جمع‌بندی:
۴	۴ نکات:
۴	۴ سؤالات تحلیلی پایانی:

۱ دستورکار شماره دو:

آشنایی با مفاهیم همزمانی، زمانبندی و مدیریت نخ‌ها در سطح کرنل

۲ اهداف آموزشی:

- آشنایی با ساختار داخلی Pintos
- اضافه کردن قابلیت توسعه سیستم زمانبندی و همزمانی Pintos

۳ توضیح پروژه:

این پروژه از چند بخش تشکیل شده است که در ادامه توضیح داده می‌شود.

۱-۱ پیش‌نیازها و ابزار مورد نیاز:

- آشنایی با زبان C و ساختارهای داده‌ی لیست و صف
- مفاهیم پایه سیستم‌عامل: نخ (Thread)، زمان‌بند (Interrupt)، وقفه (Schedule)
- محیط نصب شده QEMU و شبیه‌ساز Pintos
- ابزارهای کامپایل (make, gcc) و اشکال‌زدایی (gdb)

۲-۱ ساختار کلی پروژه :Pintos

- کد پروژه در مسیر pintos/src قرار دارد.
- پوشه threads شامل کدهای مربوط به کرنل چندنخی است که در این آزمایش تغییر داده می‌شود.
- پوشه devices شامل درایورهای سخت‌افزاری از جمله timer است.
- کامپایل با دستور make و اجرای تست‌ها با pintos --qemu -v -- run test_name انجام می‌شود.

۳-۱ مراحل انجام آزمایش:

۱-۱-۱ فاز اول - آماده‌سازی و شناخت ساختار

- اجرای پروژه‌ی اولیه Pintos
- مرور فایل‌های timer.c, synch.c, thread.c
- بررسی روند schedule و switch_threads با استفاده از gdb
- توجه به محدودیت ۴ KB حافظه پشته هر نخ

۱-۱-۲ فاز دوم - طراحی:

- تکمیل فایل DESIGNDOC در پوشه threads
- مستندسازی ساختارهای داده، الگوریتم‌ها و روش‌های هم‌زمانی مورد استفاده

۱-۱-۳ فاز سوم - ساعت زنگ‌دار (Alarm Clock)

- بازنویسی تابع timer_sleep برای جلوگیری از busy-wait
- نگهداری لیستی از نخ‌های درحال خواب و زمان بیداری آنها

- بیدارسازی نخها در وقفه‌ی تایمر (timer_interrupt)

۴-۳-۳ فاز چهارم - زمان‌بند اولویت‌محور (Priority Scheduler)

- تعریف بازه‌ی اولویت از PRI_MIN تا PRI_MAX
- پیاده‌سازی preemption هنگام آماده شدن نخ با اولویت بالاتر
- بیدار شدن نخ با بالاترین اولویت هنگام آزاد کردن قفل‌ها یا semaphore
- افزودن priority donation برای حل مشکل وارونگی اولویت
- اصلاح قفل‌ها برای پشتیبانی از donation های تودرتو
- پیاده‌سازی thread_get_priority و thread_set_priority

۵-۳-۳ فاز پنجم - زمان‌بند پیشرفته (Advanced Scheduler - MLFQS)

- پیاده‌سازی متغیرهای load_avg و recent_cpu، nice
- بهروزرسانی اولویت‌ها بر اساس فرمول BSD Scheduler
- غیرفعال شدن priority دستی در حالت MLFQS
- استفاده از حساب اعشاری ثابت (Fixed-Point Arithmetic) برای محاسبات

۶-۳-۳ فاز ششم - تست و ارزیابی:

- اجرای تست‌ها با دستور make check
- بررسی نتایج تست‌ها در مسیر tests/threads
- اطمینان از عبور تست‌های mlfqs و alarm, priority

۷-۳-۳ جمع‌بندی:

به طور خلاصه برخی از فایل‌هایی که تغییر می‌یابند در جدول ۱ ذکر شده‌اند.

جدول ۱ - برخی از فایل‌هایی که می‌باید تغییر یابند

نقش	فایل
timer_sleep	devices/timer.c
اضافه کردن زمان‌بندها و priority	threads/thread.c
تعریف ساختار thread و اولویت‌ها	threads/thread.h
تغییر برای donation	threads/synch.c
پیاده‌سازی محاسبات اعشاری ثابت	threads/fixed-point.h

همچنین جدول ۲ جمع‌بندی فازها را نشان می‌دهد.

جدول ۲ - خلاصه وظایف فازها

فاز	عنوان	فایل‌های کلیدی	خروجی مورد انتظار
۱	آماده‌سازی و شناخت ساختار	thread.c , synch.c	درک ساختار پایه
۲	طراحی	DESIGNDOC	طراحی مستند سیستم
۳	Alarm Clock	timer.c	حذف busy, خواب صحیح نخها
۴	Priority Scheduler	thread.c , synch.c	زمانبندی بر اساس اولویت و donation
۵	(MLFQS) پیشرفتی زمانبند پویا	thread.c, fixed-point.h	پیاده‌سازی زمانبند پویا
۶	تست و ارزیابی	-	انجام تست‌ها و مستندسازی

۴-۳ نکات:

- از تعریف آرایه‌های بزرگ به صورت محلی در پشته خودداری شود.
- تغییرات فقط در فایل‌های threads و devices اعمال شود.
- کد تمیز، دارای توضیح و هماهنگ با سبک Pintos باشد.

۴ سوالات تحلیلی پایانی:

- (۱) تفاوت بین روش timer_sleep و روش busy-wait block در پیاده‌سازی چیست؟
- (۲) در شرایطی که دو نخ با اولویت متفاوت برای یک قفل رقابت می‌کنند، donation چگونه عمل می‌کند؟
- (۳) در سیستم MLFQS، پارامتر nice چه نقشی در کاهش یا افزایش اولویت دارد؟
- (۴) تفاوت اصلی بین زمانبند ساده و MLFQS در پاسخ‌گویی به بار سیستم چیست؟
- (۵) چرا استفاده از floating-point arithmetic داده می‌شود؟

تحویل پروژه:

- گزارشکار به همراه سورس کدهای خود را پوشه‌ای با نام osLab_P1_stdID ارسال کنید.
- مهلت ارسال یکشنبه ۱۴۰۴/۰۸/۲۵ ساعت ۱۳ می‌باشد.
- ارائه حضوری یکشنبه ۱۴۰۴/۰۸/۲۵ در زمان برگزاری کلاس خواهد بود.

موفق باشید - آهوز