

به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

سیستم‌های عامل

گزارش پروژه (فاز سوم)

پرهام احمدی - ۹۸۳۱۰۷۱

بردیا اردکانیان - ۹۸۳۱۰۷۲

- بخش اول : پیاده‌سازی الگوریتم‌های زمان‌بندی..... ۲
- الگوریتم Round Robin..... ۲
- الگوریتم Preemptive Priority..... ۲
- الگوریتم Preemptive Multi Layered Priority Queue..... ۲
- الگوریتم Lottery..... ۲
- بخش دوم : کدهای کمکی یا ارزیابی..... ۲
- قابلیت تغییر سیاست زمان‌بندی..... ۳
- قابلیت اندازه‌گیری زمان..... ۳

بخش اول : پیاده سازی الگوریتم های زمان بندی

در این گزارش از توضیحات مربوط به نحوه ایجاد فراخوانی سیستمی اشاره نشده است.

الگوریتم Round Robin

برای پیاده سازی این الگوریتم در ابتدا مطابق دستور پروژه متغیری برای تعیین مقدار کوآنتوم تعریف شده است و سپس در تابع scheduler و همچنین تابع trap تغییراتی برای اعمال مقدار کوآنتوم و محاسبه زمان خروج پردازش از پردازنده انجام شده است.

جهت ارزیابی این الگوریتم تستی به نام فایل roundRobinTest.c نوشته شده است.

با تغییر مقدار کوآنتوم به این نتیجه می رسیم که اگر کوآنتوم را بیش از حد کم قرار دهیم waiting time کاهش می یابد ولی زمان بیشتری را صرف context switch می کنیم و turnaround time افزایش می یابد ولی با افزایش مقدار کوآنتوم تا یک حدی زمان پاسخ کاهش می یابد و از آن حد به بعد دوباره افزایش می یابد.

الگوریتم Preemptive Priority

در پیاده سازی این الگوریتم ابتدا پردازشها بر اساس اولویت مرتب می شوند و از پردازشهای دارای اولویت یکسان صفی تشکیل می شود که به صورت round robin اجرا می شوند و پس از اتمام این صف دوباره برای پردازشهایی که اولویت کمتر دارند این صف تشکیل می شود. و همچنین جهت preemption در تابع trap در هر کلاک کل پردازشها یک دور بررسی می شوند و در صورت وجود پردازشهای با اولویت بیشتر پردازش قبلی از حالت اجرا در می آید. جهت تعیین اولویت پردازش فراخوانی سیستمی setPriority تعریف شده است.

جهت ارزیابی این الگوریتم فایل تستی به نام prioritySchedTest.c نوشته شده است.

الگوریتم Preemptive Multi Layered Priority Queue

پیاده سازی این الگوریتم در تمام جزئیات مانند حالت قبل است به جز کوآنتومی که بر اساس اولویت به هر پردازش اختصاص پیدا می کند.

جهت ارزیابی این الگوریتم فایل تستی به نام multiLayeredQueuedTest.c نوشته شده است.

الگوریتم Lottery

استفاده از این الگوریتم شرعا و قانونا حرام است.

این الگوریتم به صورت شانسی کار می کند. به این صورت که در هر مرحله یک عدد رندوم را ایجاد و پردازشهای را که بلیت آن بیشتر از آن عدد باشد اجرا می کند. (همانند لاتاری) بلیت هر پردازش به صورت پیش فرض ۱۰ قرار داده شده است و با استفاده از فراخوانی سیستمی setTickets می توان مقدار بلیت را تغییر داد تا شانس پردازش را برای اجرا شدن بیشتر کند.

جهت ارزیابی این الگوریتم فایل تستی به نام lotteryTest.c نوشته شده است.

بخش دوم : کدهای کمکی یا ارزیابی

قابلیت تغییر سیاست زمان بندی

به منظور تغییر سیاست زمان بندی فراخوانی سیستمی با نام `setSchedulerStrategy` تعریف شده است که ورودی های آن به این صورت است :

- ۰- الگوریتم پیش فرض
- ۱- الگوریتم round robin
- ۲- الگوریتم priority
- ۳- الگوریتم layered queue
- ۴- لاتاری

قابلیت اندازه گیری زمان

برای اندازه گیری زمان فراخوانی های سیستمی زیر تعریف شده است:

- `getTerminateTime` (برگرداننده زمان پایان پردازش)
- `getTurnaroundTime`
- `getWaitingTime`
- `getburstTime`
- `getEnteringTime` (برگرداننده زمان ورود پردازش به صف اجرا)