



سیستم‌های عامل - نیمسال اول ۱۴۰۱-۱۴۰۰

پروژه پایانی - موعده تحویل: ۱۴۰۰/۱۰/۱۲

شرح پروژه درس: در این بخش از درس، دانشجویان می‌توانند در قالب گروه‌های یک یا دو نفره نسبت به انتخاب و پیاده‌سازی یکی از دو پروژه مندرج در این مستند، اقدام نمایند. در انجام پروژه درس، نکات زیر باید مورد توجه قرار گیرد:

- هر یک از اعضای گروه‌ها، پروژه خود را به صورت جداگانه در سامانه آموزش الکترونیک، بارگذاری و ارسال نمایند.
- پیاده‌سازی پروژه، می‌تواند به یکی از زبان‌های Perl، Python، C++ یا Java صورت گیرد و لازم است فایل‌های مربوط به پیاده‌سازی نیز و نسخه قابل اجرای پروژه، ارسال شود.
- فایل ورودی و خروجی برنامه، نیز بایستی جزء اطلاعات ارسالی باشد.
- همچنین به انضمام موارد فوق، ضروری است یک گزارش (حداقل ۶ صفحه و حداکثر ۱۰ صفحه) نیز تهیه و ارسال شود که طی آن، به طور مختصر و مفید، جزئیات پیاده‌سازی پروژه انجام شده، مطرح گردد و نتایج حاصل به صورت تحلیلی، مورد بحث و بررسی قرار گیرد.

نحوه ارسال: صرفاً از طریق صفحه درس در سامانه آموزش الکترونیک دانشگاه تهران. لطفاً از ارسال تمرین‌های این درس با استفاده از ایمیل، جداً خودداری نمایید.

موضوع پروژه شماره یک: شبیه‌سازی الگوریتم‌های زمان‌بندی واحد پردازشگر مرکزی (CPU Scheduling)

هدف پروژه شماره یک: هدف از انجام این پروژه، شبیه‌سازی الگوریتم‌های مختلف زمان‌بندی است که سیستم عامل از آنها برای زمان‌بندی CPU و تخصیص آن به پردازش‌های مختلف، استفاده می‌کند. همچنین با استفاده از نتایج این شبیه‌سازی، دانشجو بایستی به صورت شهودی، تفاوت بین این الگوریتم‌ها را دریابد و قادر باشد آنها را با یکدیگر مقایسه کند.

شرح پروژه شماره یک: برای هر پردازش پارامترهای زیر، مد نظر می‌باشد:

- شناسه پردازش (Process ID)
- زمان ورود به سیستم (Arrival Time)
- اولویت (Priority)
- تعداد واحد زمانی مورد نیاز از CPU. در این پروژه، هر واحد زمانی، مدت زمانی است که CPU، قطعه کد زیر را اجرا می‌کند:

```
for (int i=0; int temp =0; i < 10000; i++)  
if (i mod 2 == 0)  
    temp= i/2  
else  
    temp=2*i;
```

این زمان‌بند شبیه‌سازی شده، هر بار پس از ورود یک پردازش (یا بند) به صف پردازش‌های آماده، اجرا می‌شود. در هر بار اجرای برنامه، کاربر باید قادر به تعیین نوع زمان‌بندی مورد استفاده باشد. این برنامه بایستی قادر به شبیه‌سازی الگوریتم‌های زمان‌بندی زیر باشد:

- FIFO
- Preemptive SJF
- Non-Preemptive SJF
- RR (with specified time quantum)
- Priority with preemption
- Non-Preemptive priority

در هر بار اجرای برنامه، ۱۰۰,۰۰۰ پردازش باید شبیه‌سازی شود. به عبارت دیگر، پس از کامل شدن اجرای ۱۰۰,۰۰۰ پردازش، اجرای برنامه بایستی متوقف شده و اطلاعات آماری زیر، محاسبه گردد و نمایش داده شود:

- تعداد پردازش‌ها: ۱۰۰,۰۰۰
- مدت زمان سپری شده (برای زمان‌بند)
- Throughput
- CPU utilization
- Average waiting time
- Average turnaround time
- Average response time

تمامی درخواست‌ها (پردازه‌ها)، بایستی از طریق یک تولید کننده درخواست (Request Generator)، تولید شده و در یک فایل ذخیره شود. این فایل به عنوان ورودی برنامه اصلی (زمان‌بند)، مورد استفاده قرار می‌گیرد. برای سهولت بیشتر، فرض کنید که برای پردازه‌ها، حالت انتظار (waiting) نداریم. همچنین فرض کنید اندازه صف پردازه‌های آماده (Ready)، برابر با ۱۰۰ باشد.

موضوع پروژه شماره دو: شبیه‌سازی الگوریتم‌های تخصیص منابع به پردازه‌ها (Resource Allocation)

هدف پروژه شماره دو: هدف از انجام این پروژه، شبیه‌سازی الگوریتم بانکداران در تخصیص منابع به پردازه‌ها و رفع بن‌بست‌های احتمالی است که سیستم عامل از آنها برای تخصیص منابع مورد نیاز به پردازه‌های مختلف، استفاده می‌کند تا ترتیب امنی از اجرای پردازه‌ها را تعیین نماید.

شرح پروژه شماره دو: دانشجویان بایستی به منظور دریافت اطلاعات مورد نیاز، واسط کاربری مناسبی طراحی کنند که ابتدا به ازای سیستم عامل، تعداد سطوح اولویت تعداد انواع منابع مختلف، تعداد نمونه‌های هر نوع منبع و نیز امکان‌پذیری استفاده اشتراکی از آن نوع منبع را از کاربر دریافت نماید. بعضی از منابع، قابلیت استفاده اشتراکی (بصورت همزمان) توسط تعدادی پردازه را دارا می‌باشند. سپس بنا به انتخاب کاربر، به ازای هر پردازه، علاوه بر مشخص کردن اولویت، تعداد کل نمونه‌های مورد نیاز از هر کدام از انواع منابع و نیز تعداد منابع تخصیص یافته تا بحال، به یکی از دو روش زیر تعیین شود:

- این کار می‌تواند بصورت تصادفی از طریق برنامه شما صورت گیرد،
 - یا اینکه مستقیماً از سوی کاربر، تعیین شود.
- در هر حالت، تعداد کل منابع تخصیص یافته از هر نوع، نباید از تعداد کل نمونه‌های موجود از آن منبع، بیشتر شود. در ادامه، برنامه شما بایستی:

- اولاً متناظر با داده‌های ارائه شده، گراف تخصیص منبع (Resource-Allocation Graph) را ترسیم نماید.
- سپس با اجرای الگوریتم بانکداران (Banker's Algorithm) و در صورت وجود، ترتیب (های) امن از اجرای پردازه‌ها مشخص و گزارش شود.
- چنانچه ترتیب امنی به ازای شرایط اولیه پردازه‌ها وجود نداشته باشد، با عنایت به اولویت اعلام شده پردازه‌ها، پردازه (های) قربانی برای جلوگیری از وقوع بن‌بست، مشخص شده و سپس بر اساس آن، ترتیب (های) امن از اجرای بقیه پردازه‌ها تعیین شود.

تندرست و پیروز باشید.

کیهانی پور