



## فاز دوم پروژه (انحصار متقابل)

در این پروژه به پیاده‌سازی الگوریتم انحصار متقابل سینگهال (مبتنی بر مهره) خواهید پرداخت.

### ۱ شرح پروژه

در این پروژه، نیاز داریم تا در محیطی توزیع شده و غیرمتمرکز، محاسباتی توسط گره‌های موجود در محیط انجام شود. عملیات پردازشی، از نوع قبضه‌نشده<sup>۱</sup> و نیازمند منابعی هستند که میان تمام گره‌های سامانه به اشتراک گذاشته شده و تعداد آنها نیز محدود است. همچنین در هر لحظه، تنها گره‌هایی مجوز انجام محاسبات را دارند که منابع مورد نیاز خود را کسب کرده باشند.

#### أ. جزئیات مساله

هر گره در این سامانه دارای شناسه منحصر به فرد است و ارتباطات میان گره‌ها از طریق مبادله پیام صورت می‌گیرد.

در این محیط توزیع شده یک گره مرکزی هماهنگ‌کننده<sup>۲</sup> وجود دارد که تنها دو وظیفه دارد:

۱. مطلع ساختن گره‌ها از وظایفشان

۲. جمع‌آوری گزارشات تولید شده توسط گره‌های محاسباتی در سامانه.

هر گره محاسباتی در ابتدا و انتهای هر وظیفه یا رخداد، گزارشی را تولید کرده و به گره مرکزی ارسال می‌کند. گزارشات جمع‌آوری شده در دسته‌های مجزا تقسیم‌بندی می‌شود و در صورت نیاز اپراتور، به او نمایش داده خواهد شد. این گزارشات شامل اطلاعاتی نظیر شروع وظیفه، اتمام وظیفه و درخواست/دریافت منبع از سایرین (با ذکر مشخصات گیرنده/فرستنده) است.

در زمان شروع به کار سامانه، هر گره از گره مرکزی وظیفه خود را جویا می‌شود. اینکار بعد از اتمام هر وظیفه تا زمانی که وظیفه دیگری برای گره مذکور وجود نداشته باشد، ادامه پیدا می‌کند. این به این معنی است که هیچ‌کدام از آنها از وظایف دیگری مطلع نیست.

#### ب. راهکار حل مساله

در کلاس درس، الگوریتم انحصار متقابل سینگهال که مبتنی بر مهره است را آموختید. در مطالب تدریس شده، این الگوریتم در شرایطی استفاده شده است که تنها یک نوع منبع و به تعداد یک واحد از آن داشته باشیم. برای حل مساله بایستی الگوریتم سینگهال را برای تعداد بیشتری از هر منبع ارتقا دهید.

<sup>۱</sup> Non-Preemptive  
<sup>۲</sup> Coordinator

الگوریتم بهبود یافته توسط شما باید مشکل بن‌بست را نیز مرتفع نماید. برای مثال، ۵ گره و ۱۰ واحد از منبع اصلی در سامانه موجود است. گره ۱، ۳ و ۴ هرکدام برای انجام محاسباتشان به ترتیب به ۴، ۵ و ۷ واحد از منبع مشترک نیاز دارند. مجموع منابع مورد نیاز توسط این ۳ گره، از حداکثر منابع سیستم بیشتر است. همانطور که در جزییات مساله ذکر شد، گره‌ها از وظایف کلی و طبعاً از نیازهای آینده خود با خبر نیستند. در چنین شرایطی احتمال وقوع بن‌بست بسیار زیاد خواهد بود. الگوریتم بهبود یافته‌ی شما باید بتواند با کمترین سربار ممکن (اعم از زمانی و تعداد بسته‌های رد و بدل شده) از وقوع بن‌بست جلوگیری کند. ارزیابی این قسمت از کار شما بر اساس نحوه مدیریت سربار توسط شما انجام خواهد شد.

## ج. بخش امتیازی

پیاده‌سازی موارد زیر امتیاز اضافه‌ای را برای شما در پی خواهد داشت:

- خط‌مشی انجام وظایف را به نحوی انتخاب کنید که میانگین زمان انجام وظایف در کل سامانه حداقل شود. در مورد رویکرد خود توضیح دهید و سپس نتایج را با خط‌مشی FIFO با استفاده از نمودار و جدول مقایسه کنید. این مقایسه باید شامل میانگین زمان انجام وظایف و میانگین زمان انتظار گره‌ها برای به‌دست آوردن منابع مورد نیاز نیز باشد. مقایسه پارامترهای جذاب دیگر نیز در ارزیابی موثر خواهند بود.

## ۲ طراحی

در این بخش قراردادهای و نکاتی که لازم است در طراحی سامانه مورد توجه قرار گیرند، ذکر شده است.

### ا. قراردادهای طراحی

- گره‌ها را نمونه‌های مختلف از یک کلاس با نام CNode در نظر بگیرید.
- هماهنگ‌کننده به آدرس دستیابی به گره‌ها در هنگام شروع کار آگاه است.
- سیاست سرویس دادن به درخواست‌ها به صورت پیش‌فرض از نوع FIFO است.
- تنها یک نوع منبع و به تعداد N واحد در سامانه موجود است.
- منابع اشتراکی سامانه در ابتدا در اختیار گره با کوچکترین شناسه است.
- در صورت نیاز فرض کنید که گره‌ها به یکدیگر در ارتباطاتشان اعتماد کامل دارند.
- تخصیص کارها به صورت دسته‌ای و در زمان شروع به کار سامانه انجام می‌شود. اپراتور وظایف گره‌ها را به صورت زیر در یک فایل متنی با نام tasks.txt مشخص می‌کند و در اختیار هماهنگ‌کننده می‌گذارد:

[Task ID]:[Node ID]:[Finish Time]:[#Resources]

برای مثال، [۱۰]:[۵]:[۳۵۰۰]:[۵] یعنی وظیفه‌ای با شناسه ۱۰، با مدت زمان انجام ۳۵۰۰ میلی‌ثانیه که نیاز به داشتن ۵ واحد منبع است به گره‌ای با شناسه ۵ اختصاص داده شده است. هماهنگ‌کننده در شروع کار، اطلاعات را از این فایل متنی خوانده و سامانه را راه‌اندازی می‌کند.

## ب. نکات طراحی

برای طراحی سامانه خود نکات زیر را مد نظر قرار دهید:

- تعداد منابع و گره‌های سامانه توسط اپراتور قابل تنظیم باشد.
- ارتباط میان گره‌ها از طریق برنامه‌نویسی سوکت انجام شود.
- زبان انتخابی برای توسعه برنامه به گونه‌ای که قابلیت‌های خواسته شده را امکان‌پذیر سازد، به دلخواه شما است. با این حال پیشنهاد می‌شود از زبان‌هایی مانند C++، Java یا Python استفاده کنید.
- در پایان کار لازم است گزارشی از انجام پروژه شامل مراحل طراحی و پیاده‌سازی و چالش‌های هر کدام نیز تهیه شده و به همراه کدها ارسال شود.

## ۳ تحویل

برای انجام و تحویل پروژه به نکات زیر توجه کنید:

- پروژه در قالب گروه‌های یک یا دو نفره قابل انجام است.
- خروجی پروژه شما **یک فایل Zip** شامل هر چیزی که برای اجرای پروژه خود به آن نیاز دارید و **یک فایل PDF** حاوی گزارش است. سپس هر دو فایل را به فرمت ZIP فشرده‌سازی کرده و در کوئرا بارگذاری نمایید.
- نام فایل ارسالی باید بر اساس فرمت زیر باشد:  
شماره دانشجویی فرد دوم-شماره دانشجویی فرد اول-AOS-P2
- مهلت ارسال پروژه ساعت ۲۳:۵۹ روز ۲۳ خرداد ۱۳۹۹ می‌باشد. سیاست تاخیر، کسر یک درصد نمره به ازای هر ساعت می‌باشد.
- در صورت مشاهده هرگونه مشابهت نامتعارف بین چند پروژه، نمره پروژه همگی آن‌ها ۱۰۰- در نظر گرفته خواهد شد.
- در زمان تحویل پروژه (که متعاقبا اعلام خواهد شد)، باید از همان پروژه‌ای که در زمان معین شده ارسال کرده‌اید، استفاده کنید. پس قبل از ارسال نهایی پروژه از اجرا شدن آن اطمینان حاصل نمایید. مسئولیت هرگونه مشکلی در این مورد متوجه خود شما خواهد بود.
- بهره‌گیری از هرگونه امتیازات ناعادلانه مانند تکه کد آماده و کتابخانه‌های مربوط به پیاده‌سازی الگوریتم اصلی پروژه که متعلق به شما نباشد، اکیدا ممنوع است.
- هرگونه سوال و ابهام احتمالی را در قسمت مشخص شده در کوئرا مطرح کنید.

با آرزوی موفقیت