

# تمرین سوم عملی – سیستم های نهفته

کیوان ایچی حق

## بخش اول: توضیح کد

- برای بحث اشتراک گذاری منابع دو الگوریتم HLP و NPP پیاده سازی شدند. برای مقدار دهی الگوریتم کافی است یکی از دو مقدار کلاس ResourceManagementAlgorithm انتخاب شود.  
برای راحتی کار با کد های داده شده، هر کلاس در فایل جداگانه قرار گرفته شد که جداگانه توضیح داده خواهند شد:
- کلاس Job  
برای پیاده سازی کلی سیستم، به دو تابع getRecourseWaiting و getRemainingSectionTime نیاز نشد!  
این کلاس چندین تغییر جزئی دارد. برای مثال دو نوع اولویت تعیین شده که متغیر initial\_priority یا به اختصار ip اولویت اولیه سمافور را دارد و متغیر priority اولویت اصلی و دینامیکی را دارد که بسته به نوع الگوریتم استفاده شده می تواند تغییر کند.  
تابع getResourceHeld آیدی سمافوری را برمی گرداند که در حال اشغال توسط scheduler است و تابع nextSectionJustStarted مشخص می کند آیا می خواهیم وارد سمافور جدید شویم یا نه.
- کلاس Task  
تغییری نکرده
- کلاس Utils  
یک Enum با نام ResourceManagementAlgorithm اضافه شده
- کلاس Taskset  
تنها تغییر این کلاس اضافه کردن تابع build\_lowest\_priority\_semaphores برای ساختن متغیر lowest\_priority\_semaphores است. همانطور که از اسم متغیر پیداست، یک دیکشنری از سمافور ها را نگه می دارد که کمترین اولویت را دارند.
- کلاس Scheduler  
تابع get\_semaphore\_priority اولویت سمافور ورودی را بر حسب الگوریتم زمانبندی برمی گرداند.  
تابع get\_active\_jobs جاب هایی که تمام نشده (not completed yet) و همچنین از زمان آنها نگذشته باشد را برمی گرداند.  
تابع build\_timeline مدت اجرای هر سمافور از هر جاب از هر تسک را نمایش میدهد.  
تابع tick بر حسب آنکه جاب فعلی در حال اجرا چه وضعیتی دارد (تمام شده و ...) سمافور ها را در صورت امکان اجرا می کند.  
در الگوریتم HLP اولویت جاب برابر اولویت کمترین سمافور خواهد شد و در الگوریتم NPP اولویت جاب برابر صفر می شود.

## بخش دوم: اجرا ها

برای مقایسه دو الگوریتم NPP و HLP از تسک کیس taskset\_2.json استفاده شده. به طور کلی الگوریتم HLP به دلیل تغییر اولویت به سمافور پیشین بهینه تر از NPP عمل میکند که صرفاً اولویت را معادل 0 میگذارد. این امر در تست کسی گفته شده نشان داده شد. در تست کیس گفته شده تسک اول تسک دوم را بلاک کرده و مانع رسیدن آن به ددلاینش میشود.

اجرا با HLP:

# تمرین سوم عملی – سیستم های نهفته

کیوان ایپچی حق

```
Timeline:
0 - 1: 1-1-1
1 - 2: 2-1-0
2 - 6: 1-1-1
6 - 21: IDLE
21 - 22: 2-2-0
22 - 30: IDLE
30 - 35: 1-2-1
35 - 41: IDLE
41 - 42: 2-3-0
42 - 50: IDLE
```

اجرا با NPP:

```
Timeline:
0 - 5: 1-1-1
5 - 6: 2-1-0
6 - 21: IDLE
21 - 22: 2-2-0
22 - 30: IDLE
30 - 35: 1-2-1
35 - 41: IDLE
41 - 42: 2-3-0
42 - 50: IDLE
```

همچنین الگوریتم HLP تعداد کمتری سمافور را اصطلاحاً بلاک کرده نسبت به NPP زیرا تسک هایی که اولویت بالاتری دارند بلاک نمیشوند بلکه اولویت آنها عوض می شود. برای توصیف دقیق تر تست کیس taskset\_3.json اجرا شده که در NPP موجب از دست رفتن ددلاین به دلیل بلاک شدن تسک با اولویت بالاتر میشود نشان داده شده:

اجرا با NPP:

```
JOB 1 of TASK 1 COMPLETED AT 5 (MET DEADLINE)
JOB 1 of TASK 4 COMPLETED AT 6 (MISSED DEADLINE)
JOB 1 of TASK 3 COMPLETED AT 10 (MET DEADLINE)
JOB 1 of TASK 2 COMPLETED AT 11 (MET DEADLINE)
JOB 1 of TASK 5 COMPLETED AT 12 (MET DEADLINE)
JOB 2 of TASK 4 COMPLETED AT 22 (MET DEADLINE)
JOB 2 of TASK 3 COMPLETED AT 26 (MET DEADLINE)
JOB 2 of TASK 2 COMPLETED AT 27 (MET DEADLINE)
JOB 2 of TASK 5 COMPLETED AT 28 (MET DEADLINE)

Timeline:
0 - 5: 1-1-1
5 - 6: 4-1-0
6 - 10: 3-1-1
10 - 11: 2-1-0
11 - 12: 5-1-0
12 - 21: IDLE
21 - 22: 4-2-0
22 - 26: 3-2-1
26 - 27: 2-2-0
27 - 28: 5-2-0
28 - 30: IDLE
```

# تمرین سوم عملی – سیستم های نهفته

کیوان ایپچی حق

اجرا با HLP:

```
Run:
JOB 1 of TASK 4 COMPLETED AT 2 (MET DEADLINE)
JOB 1 of TASK 1 COMPLETED AT 6 (MET DEADLINE)
JOB 1 of TASK 3 COMPLETED AT 10 (MET DEADLINE)
JOB 1 of TASK 2 COMPLETED AT 11 (MET DEADLINE)
JOB 1 of TASK 5 COMPLETED AT 12 (MET DEADLINE)
JOB 2 of TASK 4 COMPLETED AT 22 (MET DEADLINE)
JOB 2 of TASK 3 COMPLETED AT 26 (MET DEADLINE)
JOB 2 of TASK 2 COMPLETED AT 27 (MET DEADLINE)
JOB 2 of TASK 5 COMPLETED AT 28 (MET DEADLINE)

Timeline:
0 - 1: 1-1-1
1 - 2: 4-1-0
2 - 6: 1-1-1
6 - 10: 3-1-1
10 - 11: 2-1-0
11 - 12: 5-1-0
12 - 21: IDLE
21 - 22: 4-2-0
22 - 26: 3-2-1
26 - 27: 2-2-0
27 - 28: 5-2-0
28 - 30: IDLE
```

## بخش سوک: مقایسه ها

- تاثیر offset  
تغییر offset به طور کلی می تواند روی نحوه زمانبندی و حتی احتمال از دست رفتن ددلاین تاثیر گذار باشد زیرا ترتیب و زمان ورود تسک ها اهمیت زیادی دارد. ولی به طور کلی همانطور که بالاتر توضیح داده شد HLC نسبت به offset کمتر تحت تاثیر قرار میگیرد تا NPP به دلیل بهینگی بیشتر آن
- تعداد ناحیه بحرانی  
در NPP کاهش یا افزایش تعداد ناحیه های بحرانی تاثیر زیادی روی عملکرد سیستم نمی گذارد. اما برای HLP هرچه تعداد ناحیه ها بیشتر شود تعداد بلاک ها هم به صورت متناظر زیاد می شود به طوری که به NPP میل می کند.
- تعداد تسک ها  
طبیعتا با افزایش تعداد تسک ها، تعداد بلاک ها هم برای هر دو الگوریتم HLP و NPP بیشتر می شود. این نسبت متناسب با مشخصات تسک ها می تواند کم و زیاد باشد اما به طور کلی اینطور است.