



## درس سیستم عامل

دکتر رضا انتظاری ملکی – دکتر وحید ازهری

---

تمرین سری چهارم

---

طراحان ..... امیرمحمد درپوش – زهرا علی زاده

تاریخ انتشار ..... ۱۴۰۳/۹/۱۸

تاریخ تحویل ..... ۱۴۰۳/۹/۲۵

## آداب نامه تمارین

- نمره هر تمرین از ۱۰۰ می باشد و بارم هر سوال روبه روی آن نوشته شده است.
- پاسخ های خود را تنها در صفحه کلاس در کوئرا آپلود کنید.
- فرمت فایل ارسالی zip و شامل cd، makefile و گزارش کار می باشد. نامگذاری فایل ارسالی باید مطابق زیر باشد:  
HW#\_StdID\_StdName.zip
- گزارش کار ارسالی شامل تکه کدها همراه با توضیحات مربوط به آن ها باشد. همچنین نتایج مربوطه به صورت تصویر همراه با توضیحات آن ها درج شوند.
- به هیچ وجه تمرینی را از دیگران کپی نکنید. در صورت مشاهده تقلب و کپی در تمرینات، نمره هر دو طرف صفر در نظر گرفته می شود.

## ۱. Automated Factory Assembly Line (۲۵)

شما مأمور طراحی و شبیه‌سازی یک خط تولید خودکار در یک کارخانه هستید که محصولات در چندین مرحله مونتاژ می‌شوند. این خط تولید شامل سه مرحله اصلی است: مونتاژ قطعات، بررسی کیفیت، و بسته‌بندی. هر مرحله توسط مجموعه‌ای از ماشین‌ها مدیریت می‌شود و هر محصول باید به ترتیب از تمامی این مراحل عبور کند. با این حال، فضای بین مراحل محدود است، بنابراین محصولات تنها در صورتی می‌توانند به مرحله بعدی منتقل شوند که فضایی در صف انتظار آن مرحله موجود باشد.

- مراحل خط تولید :

۱. مونتاژ قطعه :

این مرحله اول است که در آن ماشین‌ها قطعات را بهم وصل میکنند تا محصول نهایی را تولید کنند. در این مرحله تنها ۴ ماشین موجود است.

۲. بررسی کیفیت :

در مرحله دوم محصول مونتاژ در مرحله قبل بررسی می‌شود تا دارای استانداردهای کارخانه باشد. در این مرحله ۳ ماشین بررسی کیفیت داریم.

۳. بسته بندی:

این مرحله نهایی است که در آن محصول برای ارسال بسته بندی میشود. در این مرحله ۲ ماشین داریم.

- صف‌های انتظار بین مراحل : بین هر دو مرحله یک صف انتظار با ظرفیت محدود وجود دارد که محصولاتی را که منتظر پردازش هستند نگه می‌دارد.

- صف انتظار بین مرحله اول و دوم حداکثر ظرفیت ۵ محصول را دارد.

- صف انتظار بین مرحله دوم و سوم حداکثر ۳ محصول را می‌تواند نگه دارد.

حال شما باید با استفاده از counting semaphores محدودیت صف‌ها و ماشین‌های موجود در هر مرحله را شبیه‌سازی کنید همچنین اطمینان حاصل کنید که مراحل به ترتیب برای یک محصول انجام می‌شوند.

( در صورت نیاز از mutex , binary semaphore هم می‌توانید استفاده کنید. )

برای شبیه‌سازی عملیات‌های کارخانه بر روی محصول از sleep استفاده کنید.

برای پیاده سازی راحتتر، محصولات را با استفاده از thread شبیه سازی کنید. به تعداد ماشین موجود در هر مرحله تعدادی thread (محصول) به صورت همزمان وارد یک مرحله می شوند و بقیه thread (محصولات) منتظر میمانند. بلافاصله با خارج شدن یک thread ، بعدی وارد آن مرحله میشود (در صورت وجود) .

فرض کنید برای شروع ۲۰ تا محصول قرار است در سیستم وجود داشته باشد.

نمونه ای از خروجی برنامه برای ۴ محصول به صورت زیر است. از شما میخواهیم پیام های مناسب برای هر مرحله بدهید که نشان دهد محصول (thread) در چه مرحله ای و چه وضعیتی است.

```
Product 1: Assembling...
Product 2: Assembling...
Product 3: Assembling...
Product 4: Assembling...
Product 3: Assembled.
Product 3: Waiting in buffer 1.
Product 2: Assembled.
Product 1: Assembled.
Product 4: Assembled.
Product 3: Quality checking...
Product 2: Waiting in buffer 1.
Product 3: Quality checked.
Product 3: Waiting in buffer 2.
Product 2: Quality checking...
Product 1: Waiting in buffer 1.
Product 3: Packaging...
Product 2: Quality checked.
Product 2: Waiting in buffer 2.
Product 1: Quality checking...
Product 4: Waiting in buffer 1.
Product 1: Quality checked.
Product 2: Packaging...
Product 1: Waiting in buffer 2.
Product 4: Quality checking...
Product 3: Packaged.
end of pipeline 3Product 4: Quality checked.
Product 1: Packaging...
Product 4: Waiting in buffer 2.
Product 2: Packaged.
end of pipeline 2Product 4: Packaging...
Product 1: Packaged.
end of pipeline 1Product 4: Packaged.
end of pipeline 4All products have been processed.
```

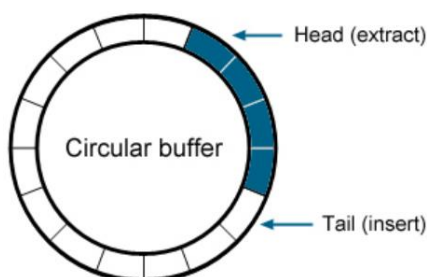
## ۲. network packet transmission. (۲۵)

در این مسئله می‌خواهیم انتقال یک بسته در شبکه اینترنت شبیه‌سازی کنیم.

در یک ارتباط اینترنتی برای انتقال یک بسته مراحل زیر طی می‌شود:

۱. فرستنده بسته خودش را که شامل یک پیام و شناسه (ID) خودش هست را آماده می‌کند و به گیرنده ارسال می‌کند.

۲. در گیرنده یک بافر حلقوی وجود دارد که دارای ۲ اشاره‌گر اصلی می‌باشد: head , tail . این اشاره‌گرها نشان می‌دهند که در حال حاضر کدام بخش از بافر اشغال شده است. وقتی یک بسته به گیرنده می‌رسد، ابتدا چک میکند که در بافر فضای خالی موجود باشد در غیر اینصورت بسته drop میشود. بافر پیاده‌سازی شده به صورت زیر است. اشاره‌گر Tail محل insert بعد و Head محل برداشت پکت برای پروسس توسط مصرف کننده می‌باشد.



۳. مصرف کننده‌هایی که در مقصد موجود هستند بسته را از محلی که Head به آن اشاره میکند، برداشته و پردازش‌های لازم را روی آن انجام می‌دهند و اشاره‌گرها را بروزرسانی می‌کنند. سپس پیام ack را برای فرستنده می‌فرستند تا اطلاع دهد که بسته را دریافت کرده است یا خیر. بدین منظور فیلد ack موجود در بسته را یک میکند تا به گیرنده اطلاع دهد که بسته دریافت شده. اگر ۱ نکند، به این معناست که بسته به دست مصرف کننده نرسیده است، پس مکانیزم تایم اوت در فرستنده فعال میشود و دیگر فرستنده منتظر ack نمی‌ماند.

۴. فرستنده بعد از ارسال بسته خود، مدت زمان معینی صبر میکند تا پیام ack را از گیرنده مبنی بر دریافت بسته، بگیرد. بعد از گذشت زمان Timeout اگر فرستنده ack را بگیرد، پیام دریافت موفقیت آمیز آن را چاپ میکند در غیر اینصورت پیامی مبنی بر عدم دریافت

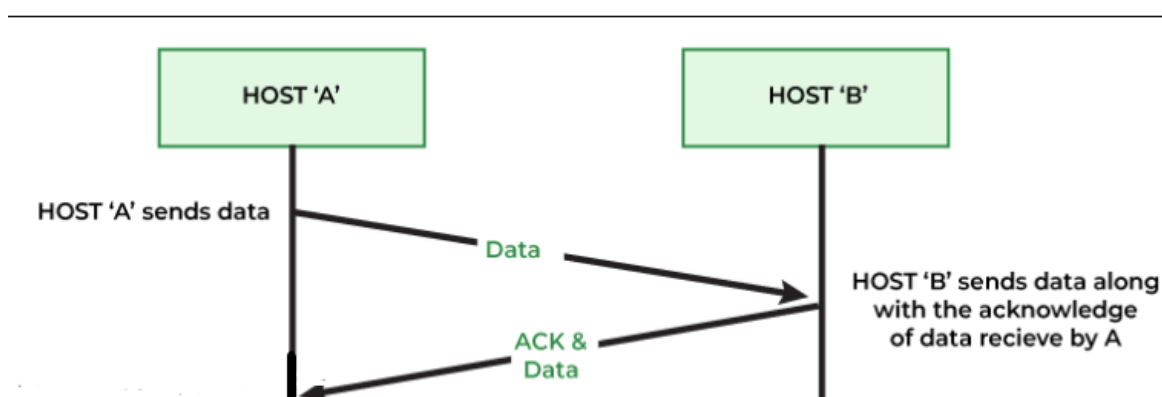
آن چاپ میکند. ( در دنیا واقعی شبکه، ارسال هر پیام ممکن است fail شود، که این مورد برای بسته ای که فرستنده تولید میکند با ack بررسی میشود و اگر خود ack در مسیر گم بشود، از پروتکل های دیگری برای اطلاع رسانی اش به فرستنده استفاده می شد که در این سوال لازم به پیاده سازی آن نیست. )

پیام ack تنها به معنی تصدیقی از سوی مصرف کننده مبنی برای دریافت بسته است که در این سوال با ۱ کردن یک فیلد در پکت شبیه سازی شده.

حال برای شبیه سازی کل این مراحل:

- از thread بعنوان تولید کننده و مصرف کننده پیام استفاده میکنیم.
- بافر حلقوی دارای محدودیت ظرفیت هست.
- دسترسی به بافر بایستی با استفاده از lock کنترل شود و thread safe باشد.
- منتظر ماندن thread ها برای مدت زمان timeout یا رسیدن ack توسط condition variable کنترل میشود.

نمونه ای از روند کلی پیاده سازی شده برای سوال :



برای شبیه سازی حالت آخر از تابع : pthread\_cond\_timedwait استفاده کنید.

نمونه ای از خروجی برنامه :

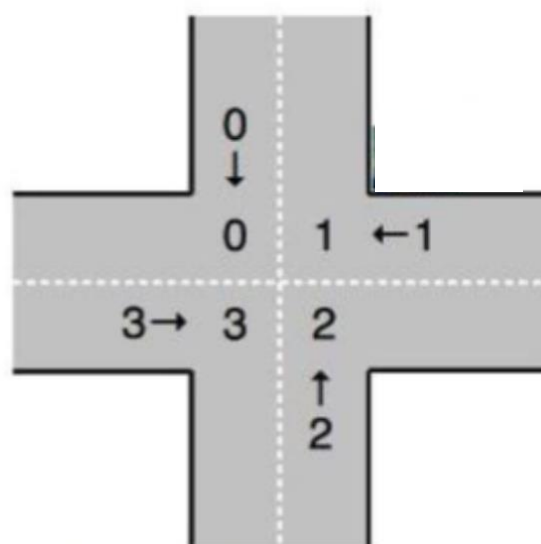
```
Sender 2 sent packet with id 777
Sender 1 sent packet with id 915
Sender 0 sent packet with id 793
Consumer consumed packet with id 777, message: Message from sender 2
Consumer sending ACK for packet id 777 to sender 2
Consumer consumed packet with id 915, message: Message from sender 1
Consumer sending ACK for packet id 915 to sender 1
Sender 2 received ACK for packet with id 777
Sender 1 received ACK for packet with id 915
Sender 2 sent packet with id 421
Sender 1 sent packet with id 362
Consumer consumed packet with id 793, message: Message from sender 0
Consumer sending ACK for packet id 793 to sender 0
Consumer consumed packet with id 421, message: Message from sender 2
Consumer sending ACK for packet id 421 to sender 2
Sender 2 received ACK for packet with id 421
Sender 0 received ACK for packet with id 793
Consumer consumed packet with id 362, message: Message from sender 1
Consumer sending ACK for packet id 362 to sender 1
Sender 2 sent packet with id 540
Sender 1 received ACK for packet with id 362
Sender 1 sent packet with id 172
Consumer consumed packet with id 540, message: Message from sender 2
Consumer sending ACK for packet id 540 to sender 2
Consumer consumed packet with id 172, message: Message from sender 1
Consumer sending ACK for packet id 172 to sender 1
Sender 2 received ACK for packet with id 540
Sender 1 received ACK for packet with id 172
Sender 0 sent packet with id 429
Sender 1 sent packet with id 782
Consumer consumed packet with id 429, message: Message from sender 0
Consumer sending ACK for packet id 429 to sender 0
Sender 0 received ACK for packet with id 429
^C
```

تمپلیت راهنما برای پیاده سازی این سوال به فایل تمرین پیوست شده است.

### ۳. Intersection Management (۲۵)

فرض کنید قرار است عبور و مرور ماشین های رباتی را در یک شهر کنترل کنیم. یک بخش مهم، عبور ماشین ها از چهار راه و عدم تصادف آن هاست. یک چهار راه در شهر ماشین های رباتی به صورت زیر است :

ترکیب چهار ربع است که از ۰ تا ۳ شماره گذاری شده اند. همچنین ۴ جهت داریم که از ۰ تا ۳ شماره گذاری شده اند. رفت و آمد ماشین ها بگونه ای است که در هر نقطه از زمان تنها یک ماشین می تواند در هر ربع باشد. در غیر این صورت تصادف به وجود می آید.



هر ماشین از یک جهت مشخص وارد چهارراه شده و از یک جهت مشخص خارج می شود. ماشین ها نمی توانند به صورت مورب حرکت کنند. به طور مثال در صورتی که ماشینی از ربع ۲ وارد چهار راه شود و مقصدش ربع ۰ باشد. باید ابتدا از ربع ۲ به ۱ و سپس به ربع ۰ برود. یا در صورتی که مقصد نهایی آن ۱ باشد تنها کافیست که از ربع ۲ به ربع ۱ برود.

راه حل فعلی تردد در این چهار راه، استفاده از حرکت غیر همزمان و نوبتی ماشین ها است. با این وجود که این راه حل امکان تصادف و ددلاک را حذف می کند، به دلیل غیرهمزمانی حرکت ماشین ها، سرعت عبور آن ها در چهار راه را نیز کاهش می دهد. به طور مثال در زمانی که یک ماشین در حال حرکت از ربع ۰ به ۳ است ماشینی دیگر می تواند به طور همزمان از ربع ۲ به ۱ حرکت کند و استفاده نکردن از این قابلیت باعث افزایش ترافیک ناخواسته در چهارراه می شود.



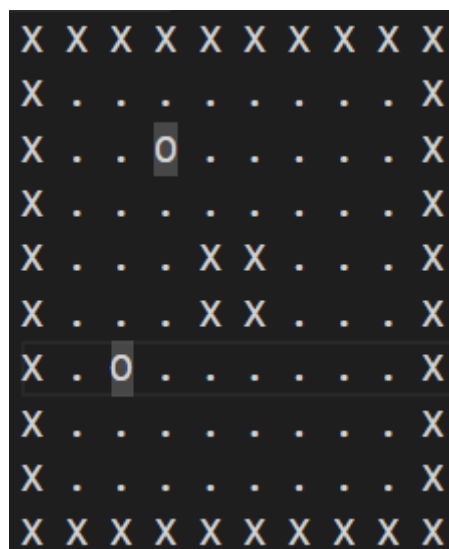
شما وظیفه دارید جوری الگوریتم همگام سازی این چهار راه را تغییر دهید که علاوه بر به وجود نیامدن ددلاک و برخورد، امکان عبور آن ها به صورت همزمان هم در چهار راه وجود داشته باشد. در فایل تمپلیت به شما کد منطق قدیمی چهار راه داده شده است و شما باید قسمت های علامت زده شده را طوری بازنویسی کنید که قابلیت حرکت همزمان ماشین ها در چهارراه موجود باشد.

## ۴. AI Agents ( ۲۵ )

در جدید ترین تحقیقات انجام شده در آزمایشگاه های هوش مصنوعی علم و صنعت، دانشجویان موفق شدند الگوریتمی برای یافتن کوتاه ترین مسیر بین دو خانه از یک جدول مستطیل شکل ارائه دهند. متأسفانه این دانشجویان در زمان انجام شبیه سازی و تست الگوریتم خود به مشکل خوردند. آن ها متوجه شدند که در صورت حرکت همزمان چند مامور در یک جدول امکان برخورد آن ها به همدیگر وجود دارد. در این سوال شما باید به دوستان خودتان در آزمایشگاه کمک کنید که مشکل برخورد مامور های خود را حل کنند.

### تعریف دقیق مسئله

در ابتدا محیط شبیه سازی، که به فرم یک جدول مستطیل شکل از علامت های مختلف است، داده می شود. هر کدام از علامت ها نشانه یک المان مشخص در صفحه شبیه سازی است.



X: دیوار

نقطه: فضای خالی

O: مامور هوش مصنوعی

در ادامه به ازای هر مامور یک موقعیت ابتدایی و یک مسیر به شما داده می‌شود. هر مامور به صورت یک رشته (thread) مجزا شبیه سازی می‌شود و می‌تواند در چهار جهت شمال، جنوب، شرق و غرب حرکت کند. (توجه کنید که مامورها تنها می‌توانند در فضا های خالی جدول حرکت کنند). همچنین بعد از حرکت مامور به خانه جدید خانه قبلی خود را خالی می‌کند. شما باید علاوه بر نوشتن منطق حرکت مامور ها روی صفحه اطمینان حاصل کنید که در طول مسیر خود به مامور دیگری برخورد نمی‌کنند.

- توجه: تمام مامور ها باید به صورت همزمان در صفحه حرکت کنند. و راه حلی که ماموران به طور نوبتی هر کدام تمام مسیر خود را طی کنند قابل قبول نیست.

## نمره دهی

برای آسانی کار، یک فایل تمپلیت برای بخش های ورودی گرفتن و نمایش صفحه شبیه سازی به شما داده می‌شود. شما باید قسمت ها علامت زده در این فایل را کامل کرده و فایل نهایی را در کوئرا آپلود کنید. برای گرفتن نمره کامل کافیهست تنها از برخورد ماموران با یک دیگر جلوگیری کنید و احتیاجی به جلوگیری از ددلاک نیست.

## نحوه ورودی

```
// board dimension and number of agents
6 3
// a NXN board
X . . . . .
. . X . . .
. . X . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . . X
// the initial position for agent 1
0 1
// path length for agent 1
8
// the path for agent 1
// each element represents a move
// R -> go Right
// L -> go Left
// U -> go Up
// D -> go Down
// S -> Stop (always comes last)
R R D D L L U S
// initial position for agent 2
4 2
// path length for agent 2
5
// path for agent 2
U R D L S
// initial position for agent 3
3 4
// path length for agent 3
5
// path for agent 3
L U R D S
```

## سوال های اختیاری

جواب به این سوالات هیچ تاثیری در نمره شما ندارد و تنها برای افراد علاقه مند می باشد.

- آیا می توان الگوریتمی ارائه داد که ددلاک را قبل از شروع شبیه سازی با توجه به کانفیگ مسئله تشخیص دهد؟
- در صورت برخورد به ددلاک در حین شبیه سازی، آیا می توانید الگوریتمی ارائه دهید که با تغییر مسیرهای مامور ها آن را برطرف کند؟
- آیا با تغییر پرایوریتی رشته (thread) مختص یک مامور در سرعت حرکت آن مامور تاثیری ایجاد می شود؟ فرض کنید کد روی یک سیستم تک هسته ای اجرا میشود.