

۴- Producer, Consumer, Controller

در این سوال قصد داریم با استفاده از حافظه اشتراکی POSIX، یک مجموعه producer، consumer و controller ایجاد کنیم.

الف) (۱۵نمره)

Producer را طوری طراحی کنید که هر چند ثانیه یک بار، یک عدد رندوم ۱ تا ۲۰ را تولید کرده و در حافظه اشتراکی قرار دهد. در صورت پر شدن حافظه، producer منتظر خالی شدن یک خانه می‌ماند. Consumer را هم طوری طراحی کنید که مقادیر را از حافظه اشتراکی برداشته و پردازش کند (پردازش خاصی مد نظر نیست). تنها هدف، ایجاد تاخیر در consumer است. میتوانید تنها از یک sleep چند ثانیه‌ای به عنوان پردازش استفاده کنید). در صورت خالی شدن حافظه، Consumer منتظر مقادیر جدید می‌ماند.

این حافظه اشتراکی به شکل stack کار می‌کند، یعنی آخرین مقادیری که Producer در حافظه قرار می‌دهد، زودتر توسط Consumer ها برداشته می‌شوند.

- پروسس‌های producer و consumer باید در فایل c جدا از هم نوشته، کامپایل و اجرا شوند. در این سوال از fork نمیتوانید برای ایجاد پروسس‌های مختلف استفاده کنید.
- کد شما باید قابلیت اجرای چند consumer همزمان با هم را داشته باشد.

برای سهولت در نمره‌دهی، لطفا نکات زیر را رعایت کنید:

- یک makefile تهیه کنید که کدهای producer و consumer را کامپایل و اجرا میکند و در گزارش سوال، دستور(های) اجرای makefileتان را بنویسید. به راهنمایی آخر سوال توجه کنید.
- از قرار دادن عدد در حافظه توسط producer و از برداشتن عدد توسط consumer، لاگ پرینت کنید تا اجرای صحیح پروسس‌ها مشخص باشد.

(ب) (۱۵ نمره)

در این قسمت باید controller را طراحی کنید. کنترلر وظیفه دارد تا consumer ها را مدیریت کند. سرعت نوشتن producer و خواندن consumer برابر نیست، بنابراین وظیفه controller این است که ظرفیت حافظه را زیر نظر داشته باشد. در صورتی که حافظه تقریباً پر شده بود، به این معناست که سرعت consumer ها به اندازه کافی زیاد نیست. بنابراین consumer جدید ایجاد می‌کند. در صورتی هم که حافظه استک تقریباً خالی شده بود، به این معناست که سرعت consumer ها خیلی زیاد است و نیازی به این تعداد پروسس consumer نیست، بنابراین consumer ها را کم می‌کند. برای تشخیص تقریباً پر بودن یا تقریباً خالی بودن حافظه، می‌توانید از یک آستانه بالا و یک آستانه پایین استفاده کنید و سعی کنید که تعداد خانه‌های پر حافظه را بین این دو آستانه نگه دارید.

- برای اجرا و از بین بردن Consumer ها، از دستورات خانواده exec و دستور kill می‌توانید استفاده کنید.
 - ظرفیت حافظه اشتراکی، تاخیر consumer و producer و آستانه بالا و پایین را می‌توانید به دلخواه خود انتخاب کنید. مهم این است که برقراری تعادل و عملکرد صحیح controller مشهود باشد. مثلاً می‌توانید ظرفیت حافظه را ۱۰۰ و تاخیر producer و consumer را به ترتیب ۳ و ۲ انتخاب کنید و سعی کنید که تعداد خانه‌های پر حافظه را بین ۱۰ تا ۲۰ نگه دارید.
- برای سهولت در نمره‌دهی، لطفاً نکات زیر را رعایت کنید:
- یک makefile تهیه کنید که کدهای producer و controller را کامپایل و اجرا می‌کند و در گزارش سوال، دستور(های) اجرای makefile تان را بنویسید.
 - از ظرفیت حافظه، ساخته شدن consumer جدید و حذف کردن consumer ها لاگ پرینت کنید تا اجرای صحیح consumer مشخص باشد.

برای نوشتن makefile ای به صورت همزمان producer و consumer یا producer و controller را اجرا کند، می‌توانید از این makefile استفاده کنید:

```
compile:
    gcc -lpthread -lrt producer.c -o producer
    gcc -lpthread -lrt consumer.c -o consumer

run: run_producer run_consumer1 run_consumer2
run_producer:
    ./producer
run_consumer1:
    sleep 5
    ./consumer
run_consumer2:
    sleep 10
    ./consumer
```

```
linser@N580VD:~/HW/POSIX/q1$ make compile
gcc -lpthread -lrt producer.c -o producer
gcc -lpthread -lrt consumer.c -o consumer
linser@N580VD:~/HW/POSIX/q1$ make -j run
```

این makefile، ابتدا producer و سپس consumerها را با تاخیر ۵ و ۱۰ ثانیه ایجاد می‌کند. آپشن -j در دستور make، برای اجرای موازی چند فایل به صورت همزمان است.