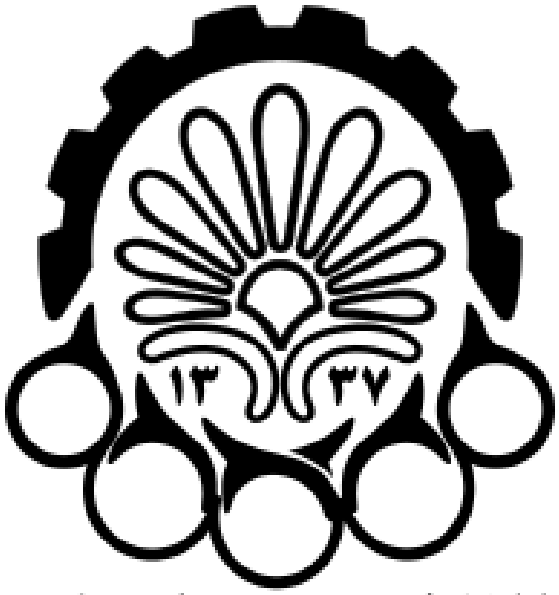

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

آزمایشگاه سیستم‌های عامل

رادین شایانفر

پاییز ۱۳۹۹

- **مسئله خوانندگان-نویسندگان:**

با دستور زیر برنامه را اجرا می‌کنیم:

```
make run_reader_writer
```

در این برنامه ۵ خواننده و یک نویسنده به طور مشترک به متغیر `count` دسترسی دارند. به دلیل وجود شرایط مسابقه برای جلوگیری از مشکل از ۲ قفل `mutex` و `rw_mutex` به صورت مشترک بین فرایندها استفاده شده است. در این برنامه اولین خواننده‌ای که وارد می‌شود قفل `rw_mutex` را می‌گیرد و آخرین خواننده‌ای که خارج می‌شود آن را آزاد می‌کند. برای شمارش تعداد خواننده‌های وارد شده نیز از متغیر `read_count` استفاده شده است که با قفل `mutex` انحصار متقابل آن تضمین می‌شود.

```
radin ~/os-lab/Lab6  master  make run_reader_writer
Reader: PID: 17199      count: 0
Reader: PID: 17200      count: 0
Reader: PID: 17199      count: 0
Reader: PID: 17201      count: 0
Reader: PID: 17200      count: 0
Reader: PID: 17202      count: 0
Reader: PID: 17199      count: 0
Reader: PID: 17203      count: 0
Writer: PID: 17198      count: 1
Reader: PID: 17201      count: 1
Reader: PID: 17200      count: 1
Reader: PID: 17202      count: 1
Reader: PID: 17199      count: 1
Reader: PID: 17203      count: 1
Writer: PID: 17198      count: 2
Reader: PID: 17201      count: 2
Reader: PID: 17200      count: 2
Reader: PID: 17199      count: 2
Reader: PID: 17202      count: 2
Reader: PID: 17203      count: 2
Writer: PID: 17198      count: 3
Reader: PID: 17201      count: 3
Reader: PID: 17200      count: 3
Reader: PID: 17199      count: 3
Reader: PID: 17202      count: 3
Reader: PID: 17203      count: 3
Writer: PID: 17198      count: 4
Reader: PID: 17203      count: 4
Reader: PID: 17201      count: 4
Reader: PID: 17200      count: 4
Reader: PID: 17199      count: 4
Reader: PID: 17202      count: 4
Writer: PID: 17198      count: 5
Reader: PID: 17201      count: 5
Reader: PID: 17203      count: 5
Reader: PID: 17200      count: 5
Reader: PID: 17199      count: 5
Reader: PID: 17202      count: 5
radin ~/os-lab/Lab6  master
```

شکل (۱) – اجرای کد مسئله‌ی خوانندگان-نویسندگان

- **مسئله فیلسوف‌های غذاخور:**

با دستور زیر برنامه را اجرا می‌کنیم:

```
make run_dining_philosophers
```

در این مسئله از ۵ قفل `chopstick[5]` برای جلوگیری از ورود هم‌زمان به ناحیه بحرانی استفاده شده است. هر فیلسوف در ترد جداگانه‌ای به تعداد بار مشخص (به طور خاص ۲ بار در شکل زیر) ابتدا فکر می‌کند، سپس ۱ ثانیه غذا می‌خورد. هر چند این روش انحصار متقابل را تضمین می‌کند اما امکان به وجود آمدن بن‌بست (در صورتی که همه‌ی فیلسوف‌ها ابتدا قفل اول خود را، پیش از این که کسی قفل دوم خود را در اختیار بگیرد، در اختیار بگیرند) در آن وجود دارد.

```
radin ~/os-lab/Lab6 master make run_dining_philosophers
Philosopher 2 is thinking.
Philosopher 4 is thinking.
Philosopher 4 is eating using chopstick[3] and chopstick[4].
Philosopher 5 is thinking.
Philosopher 3 is thinking.
Philosopher 2 is eating using chopstick[1] and chopstick[2].
Philosopher 1 is thinking.
Philosopher 4 finished eating.
Philosopher 4 is thinking.
Philosopher 1 is eating using chopstick[0] and chopstick[1].
Philosopher 2 finished eating.
Philosopher 2 is thinking.
Philosopher 1 finished eating.
Philosopher 1 is thinking.
Philosopher 1 is eating using chopstick[0] and chopstick[1].
Philosopher 1 finished eating.
Philosopher 5 is eating using chopstick[4] and chopstick[0].
Philosopher 5 finished eating.
Philosopher 5 is thinking.
Philosopher 5 is eating using chopstick[4] and chopstick[0].
Philosopher 5 finished eating.
Philosopher 4 is eating using chopstick[3] and chopstick[4].
Philosopher 4 finished eating.
Philosopher 3 is eating using chopstick[2] and chopstick[3].
Philosopher 3 finished eating.
Philosopher 3 is thinking.
Philosopher 3 is eating using chopstick[2] and chopstick[3].
Philosopher 3 finished eating.
Philosopher 2 is eating using chopstick[1] and chopstick[2].
Philosopher 2 finished eating.
radin ~/os-lab/Lab6 master
```

شکل (۲) – اجرای کد مسئله‌ی فیلسوف‌های غذاخور