



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

آزمایشگاه سیستم‌های عامل

رادین شایانفر

پاییز ۱۳۹۹



آزمایش پنجم

ابتدا با رسم هیستوگرام در ۲ حالت سریال و هم‌رند به ازای نمونه‌ی ۱۰۰ تایی، درستی کد را بررسی می‌کنیم.

```
radin ~/os-lab/Labs master make serial && ./serial.out 100

*
***
*****
*****
*****
*****
*****
*****

radin ~/os-lab/Labs master make concurrent && ./concurrent.out 100

*****
*****
*****
*****
*****
*****
*****

radin ~/os-lab/Labs master
radin ~/os-lab/Labs master
```

شکل (۱) – نمودار هیستوگرام در ۲ حالت به ازای نمونه‌ی ۱۰۰ تایی

(۱) زمان اجرای برنامه در حالت سریال به ازای تعداد نمونه‌های مختلف:

۲۰۰۰۰۰۰	۵۰۰۰۰	۵۰۰۰	۵۰۰	تعداد نمونه
4150ms	120ms	20ms	7ms	زمان اجرا

(۲) زمان اجرای برنامه در حالت هم‌رند به ازای تعداد نمونه‌های مختلف:

۲۰۰۰۰۰۰	۵۰۰۰۰	۵۰۰۰	۵۰۰	تعداد نمونه
1152ms	45ms	6ms	2ms	زمان اجرا



```
radin ~/os-lab/Lab5 master make serial
radin ~/os-lab/Lab5 master time ./serial.out 5000
./serial.out 5000 0.00s user 0.00s system 93% cpu 0.007 total
radin ~/os-lab/Lab5 master time ./serial.out 50000
./serial.out 50000 0.02s user 0.00s system 97% cpu 0.020 total
radin ~/os-lab/Lab5 master time ./serial.out 500000
./serial.out 500000 0.12s user 0.00s system 99% cpu 0.118 total
radin ~/os-lab/Lab5 master time ./serial.out 20000000
./serial.out 20000000 4.15s user 0.00s system 99% cpu 4.150 total
radin ~/os-lab/Lab5 master
```

شکل (۲) - زمان‌های اجرا در حالت سریال

```
radin ~/os-lab/Lab5 master make concurrent
radin ~/os-lab/Lab5 master time ./concurrent.out 5000
./concurrent.out 5000 0.00s user 0.00s system 215% cpu 0.002 total
radin ~/os-lab/Lab5 master time ./concurrent.out 50000
./concurrent.out 50000 0.02s user 0.00s system 384% cpu 0.006 total
radin ~/os-lab/Lab5 master time ./concurrent.out 500000
./concurrent.out 500000 0.20s user 0.00s system 447% cpu 0.045 total
radin ~/os-lab/Lab5 master time ./concurrent.out 20000000
./concurrent.out 20000000 4.99s user 0.00s system 433% cpu 1.152 total
radin ~/os-lab/Lab5 master
```

شکل (۲) - زمان‌های اجرا در حالت هم‌روند



آزمایش پنجم

۳) بله. متغیر `hist[counter + 12]` می‌تواند ایجاد `race condition` کند. برای جلوگیری از این اتفاق می‌توان با استفاده از `spin lock` ها و یا سمافور، انحصار متقابل در حین دسترسی به آرایه‌ی `hist` ایجاد کرد. به عنوان مثال می‌توان به ازای هر خانه‌ی `hist` یک سمافور با مقدار اولیه یک ایجاد کرد و در هنگام ایجاد تغییر در آن، به شکل زیر کد را بازنویسی کرد:

```
sem_wait(&sem[counter + 12]);  
hist[counter + 12]++;  
sem_post(&sem[counter + 12]);
```

۴) میزان افزایش سرعت در حالت هم‌راند:

تعداد نمونه	۵۰۰۰	۵۰۰۰۰	۵۰۰۰۰۰	۲۰۰۰۰۰۰۰
افزایش سرعت	5ms = 71%	14ms = 70%	75ms = 62%	2998ms = 72%