

گزارش سوال اول تمرین سری ۵ سیستم عامل

محمدپویا تراشی - ۴۰۱۵۲۱۱۰۲

دو تابع عمومی برای تمامی پردها در این سوال وجود دارد. تابع calculateSum به کمک isPrime، اول بودن اعداد یک بازه را بررسی می‌کند و آن‌ها را جمع می‌کند.

```
9  int isPrime(long num)
10 {
11     if (num < 2) return 0;
12     if (num == 2) return 1;
13     if (num % 2 == 0) return 0;
14
15     long sqrtNum = sqrt(num);
16     for (long i = 3; i <= sqrtNum; i += 2)
17     {
18         if (num % i == 0) return 0;
19     }
20     return 1;
21 }
22
23 long calculateSum(long start, long end) {
24     long sum = 0;
25     for (long i = start; i <= end; i++)
26     {
27         if (isPrime(i) == 1)
28             sum += i;
29     }
30
31     return sum;
32 }
```

در ادامه و در بدنه تابع اصلی، تعداد واحدهای پردازش منطقی موجود در سیستم به دست می‌آید و مولفه طول بازه بر این اساس محاسبه می‌شود. این مولفه مشخص می‌کند هر پردها، چه تعدادی عدد را بررسی خواهد نمود. همچنین تعداد پردها معادل تعداد هسته‌های موجود است. همچنین به تعداد هسته‌ها pipe ایجاد می‌شود که پردها داده خود را در آن ذخیره می‌کنند. این اتفاق به وسیله سیستم کال pipe رخ می‌دهد.

```
35     long nproc = sysconf(_SC_NPROCESSORS_ONLN);
36     long interval = (MAX_NUM / nproc);
37
38     int pipes[nproc][2];
39     pid_t pids[nproc];
40
41     for (int i = 0; i < nproc; i++)
42         pipe(pipes[i]);
```

در ادامه به ازای هر هسته سیستم یک پردازشگر تعریف می‌شود و هر یک عملیات زیر را انجام می‌دهند:

۱. طرف read از pipe بسته می‌شود.
۲. بازه کاری پردازشگر مربوطه محاسبه می‌شود و جمع آن حساب می‌شود.
۳. مقدار محاسبه شده در pipe مورد نظر نوشته می‌شود.
۴. بعد از نوشته شدن داده در pipe طرف write لوله نیز بسته می‌شود.

```
45     for (int i = 0; i < nproc; i++)
46     {
47         pids[i] = fork();
48
49         if (pids[i] == 0)
50         {
51             close(pipes[i][0]);
52
53             long start = i * interval + 1;
54             long end = (i + 1) * interval;
55             long intervalSum = calculateSum(start, end);
56             write(pipes[i][1], &intervalSum, sizeof(intervalSum));
57
58             close(pipes[i][1]);
59             exit(0);
60         }
61         else
62             close(pipes[i][1]);
63     }
64 }
```

در انتها پردازشگر پدر تمامی مقداری موجود در pipe را read نموده و جمع کل را محاسبه می‌نماید.

```
69     for (int i = 0; i < nproc; i++)
70     {
71         read(pipes[i][0], &intervalSum, sizeof(intervalSum));
72         total_sum += intervalSum;
73
74         printf("sum of interval %d: %ld\n", i, intervalSum);
75         close(pipes[i][0]);
76
77         waitpid(pids[i], NULL, 0);
78     }
```