

Programowanie współbieżne

Laboratorium 5 - Łącza nienazwane i nazwane

Autor: Tomasz Adamczyk

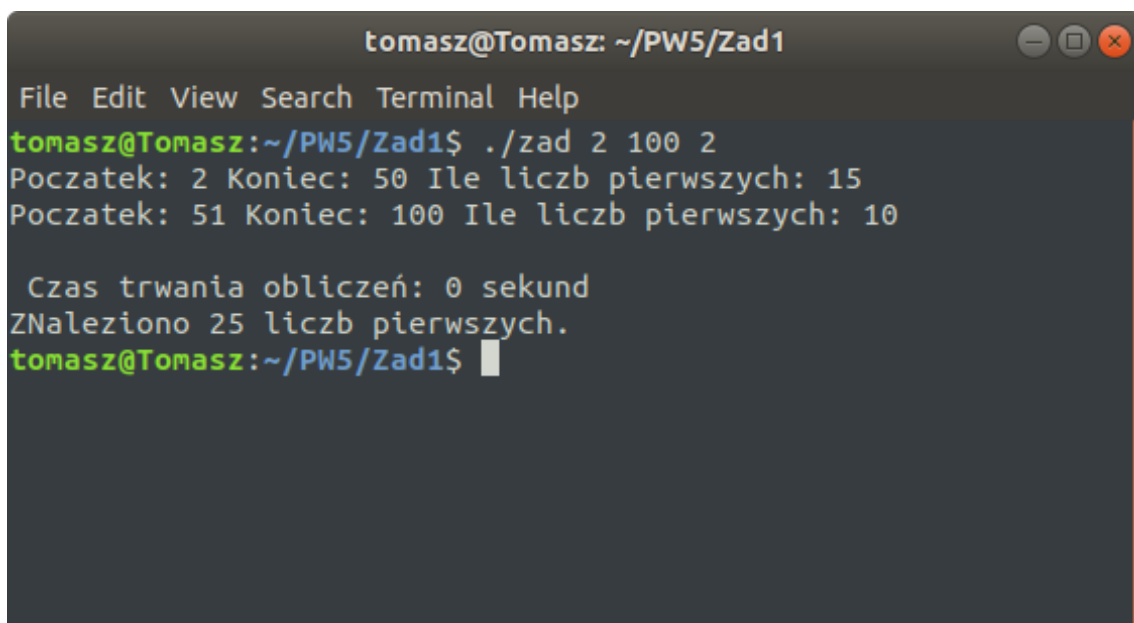
Prowadzący: Dr inż. Jędrzej Ułasiewicz

1 Zrealizowane zadania

- Znajdowanie liczb pierwszych, komunikacja poprzez łącze nienazwane
- Znajdowanie liczb pierwszych - wykorzystanie kolejki FIFO.

2 Znajdowanie liczb pierwszych, komunikacja poprzez łącze nienazwane

2.1 Wynik działania programu



```
tomasz@Tomasz: ~/PW5/Zad1
File Edit View Search Terminal Help
tomasz@Tomasz:~/PW5/Zad1$ ./zad 2 100 2
Początek: 2 Koniec: 50 Ile liczb pierwszych: 15
Początek: 51 Koniec: 100 Ile liczb pierwszych: 10

Czas trwania obliczeń: 0 sekund
ZNaleziono 25 liczb pierwszych.
tomasz@Tomasz:~/PW5/Zad1$
```

Rysunek 1: Wynik działania programu.

2.2 Kod programu

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/wait.h>
4 #include <time.h>
5 #include <errno.h>
6 #include <fcntl.h>
7 #include <stdlib.h>
8
9 struct result {
10     int start;
11     int end;
```

```

12     int howMany;
13 } ;
14
15 int isPrime(int n)
16 {
17     int i,j=0;
18     for(i=2;i*i<=n;i++)
19     {
20         if(n%i == 0) return(0) ;
21     }
22     return(1);
23 }
24
25 int main(int argc, char *argv[])
26 {
27     int pid, status, i, start, end, N, last, sum, startPart, endPart, wr, res;
28     int fd[2];
29     pipe(fd);
30     start = atoi(argv[1]);
31     end = atoi(argv[2]);
32     N = atoi(argv[3]);
33     double partition2 = (double)(end - start)/N;
34     int partition = (end - start)/N;
35     time_t timeCount;
36
37     if(partition2 > partition)
38     {
39         partition++;
40     }
41     char buffer1[10], buffer2[10], buffer3[10];
42
43     startPart = start;
44     endPart = end;
45
46     timeCount = time(NULL);
47     for(i=0; i<N; i++)
48     {
49         start = (i*partition) + startPart;
50         end = ((i+1)*partition) + startPart - 1;
51
52         if (end == endPart - 1) {
53             end = endPart;
54         }
55
56         if(fork() == 0)
57         {
58             close(fd[0]);
59             int counter = 0, primary, j;
60             struct result r;
61
62             for (j = start; j <= end; j++) {
63
64                 primary = isPrime(j);
65
66                 if (primary == 1) {
67                     counter++;
68                 }
69             }
70
71             r.start = start;
72             r.end = end;
73             r.howMany = counter;
74             wr = write(fd[1], &r, sizeof(r));

```

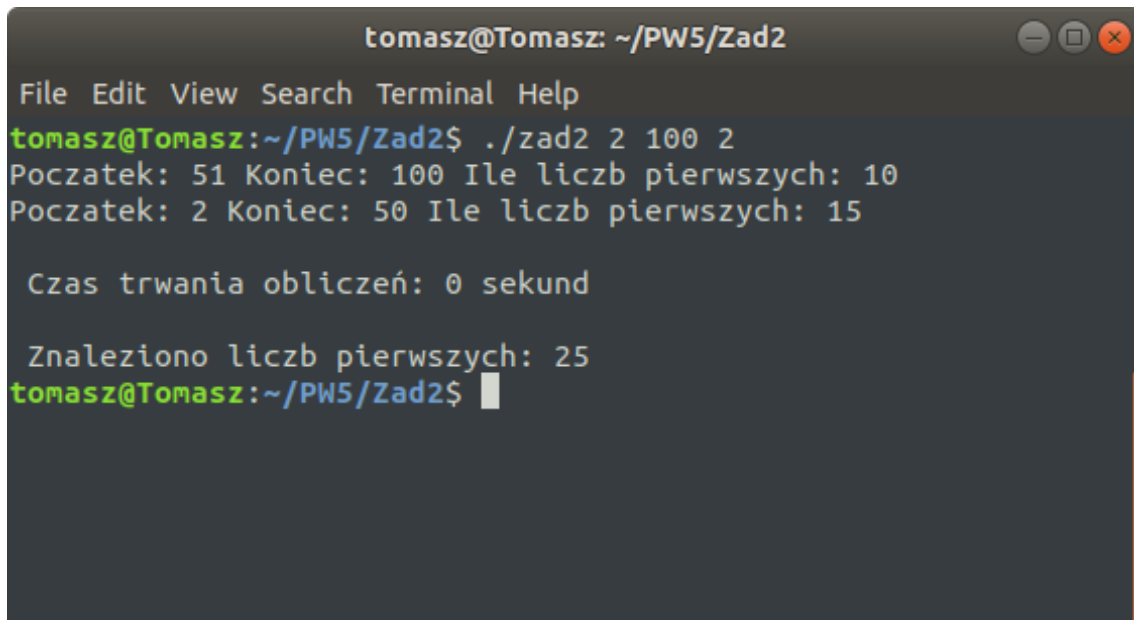
```

75     close(fd[1]);
76     exit(i);
77
78 }
79 }
80
81 close(fd[1]);
82 struct result finalResult;
83
84 for (i=0; i<N; i++) {
85     res = read(fd[0], &finalResult, sizeof(finalResult));
86     if (res < 0) {
87         perror("Bład odczytu!");
88         exit(1);
89     }
90     sum = sum +finalResult.howMany;
91     printf("Początek: %d Koniec: %d Ile liczb pierwszych: %d\n", finalResult.
start, finalResult.end, finalResult.howMany);
92
93 }
94
95 timeCount = time(NULL) - timeCount;
96
97 printf("\n Czas trwania oblicze : %ld sekund \n", timeCount);
98 printf("ZNaleziono %d liczb pierwszych. \n", sum);
99 close(fd[0]);
100
101 return 0;
102 }
103

```

3 Znajdowanie liczb pierwszych - wykorzystanie kolejki FIFO.

3.1 Wynik działania programu



```

tomasz@Tomasz: ~/PW5/Zad2
File Edit View Search Terminal Help
tomasz@Tomasz:~/PW5/Zad2$ ./zad2 2 100 2
Początek: 51 Koniec: 100 Ile liczb pierwszych: 10
Początek: 2 Koniec: 50 Ile liczb pierwszych: 15

Czas trwania obliczeń: 0 sekund

Znaleziono liczb pierwszych: 25
tomasz@Tomasz:~/PW5/Zad2$

```

Rysunek 2: Wynik działania programu.

3.2 Główny kod programu

```

1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/wait.h>
4 #include <time.h>
5 #include <errno.h>
6 #include <fcntl.h>
7 #include <stdlib.h>
8
9 struct result{
10     int start;
11     int end;
12     int howMany;
13 };
14
15 int main(int argc, char *argv[])
16 {
17     int pid, status, i, start, end, N, last, fd_read, fd, res;
18     start = atoi(argv[1]);
19     end = atoi(argv[2]);
20     N = atoi(argv[3]);
21     double partition2 = (double)(end - start)/N;
22     int partition = (end - start)/N;
23     time_t timeCount;
24
25     if(partition2 > partition)
26     {
27         partition++;
28     }
29     char buffer1[10], buffer2[10], buffer3[10];
30
31     if (mkfifo("fifo", 0666) < 0)
32     {
33         perror("mkfifo");
34         exit(0);
35     }
36
37     fd = open("fifo", O_RDWR);
38
39     timeCount = time(NULL);
40     for(i=0; i<N; i++)
41     {
42         if(fork() == 0)
43         {
44             sprintf(buffer1, "%d", i);
45             sprintf(buffer2, "%d", (i*partition)+start);
46             last = ((i+1)*partition)+start - 1;
47             if(last == end - 1)
48             {
49                 last=end;
50             }
51             sprintf(buffer3, "%d", last);
52             execl("./licz", "licz", buffer1, buffer2, buffer3, NULL);
53             exit(i+1);
54         }
55     }
56
57     for(i=0; i<N; i++)
58     {
59         pid = wait(&status);
60         //printf("Proces %d zako czony status: %d\n", pid, status);
61     }
62
63     timeCount = time(NULL) - timeCount;

```

```

64
65     int sum = 0;
66     struct result r;
67     for (int i = 0; i < N; i++) {
68
69         res = read(fd, &r, sizeof(r));
70
71         if(res<0)
72         {
73             perror("Bład odczytu");
74             exit(0);
75         }
76
77         sum = sum + r.howMany;
78         printf("Początek: %d Koniec: %d Ile liczb pierwszych: %d\n", r.start, r.end,
79             r.howMany);
80     }
81
82     printf("\n Czas trwania oblicze : %ld sekund \n", timeCount);
83     printf("\n Znaleziono liczb pierwszych: %d \n", sum);
84     close(fd);
85     return 0;
86 }

```

3.3 Kod programu wyznaczającego liczby pierwsze

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <errno.h>
5  #include <fcntl.h>
6  #include <sys/wait.h>
7  #include <sys/types.h>
8
9  struct result {
10     int start;
11     int end;
12     int howMany;
13 };
14
15 int isPrime(int n)
16 {
17     int i,j=0;
18     for(i=2;i*i<=n;i++)
19     {
20         if(n%i == 0) return(0) ;
21     }
22     return(1);
23 }
24
25 int main(int argc, char *argv[])
26 {
27     int primary;
28     int i, j, start, end, counter=0;
29     int file;
30     start = atoi(argv[2]);
31     end = atoi(argv[3]);
32     for(i=start; i<=end; i++)
33     {
34         primary = isPrime(i);
35
36         if(primary == 1)
37         {

```

```

38     counter++;
39 }
40 }
41
42 struct result r;
43 r.start = start;
44 r.end = end;
45 r.howMany = counter;
46
47 int fd = open("fifo", O_RDWR);
48 if (fd < 0) {
49     perror("Blad otwarcia.");
50     exit(0);
51 }
52
53 int status = write(fd, &r, sizeof(r));
54 if (status < 0) {
55     perror("Blad zapisu.");
56     exit(0);
57 }
58 close(fd);
59
60 return counter;
61 }
62
63

```