Sistemas Distribuídos

COS470 2016/1

Trabalho Prático 1

Débora Barbosa Pina
Patrícia de Andrade Kovaleski

Decisões de Projeto e Implementação

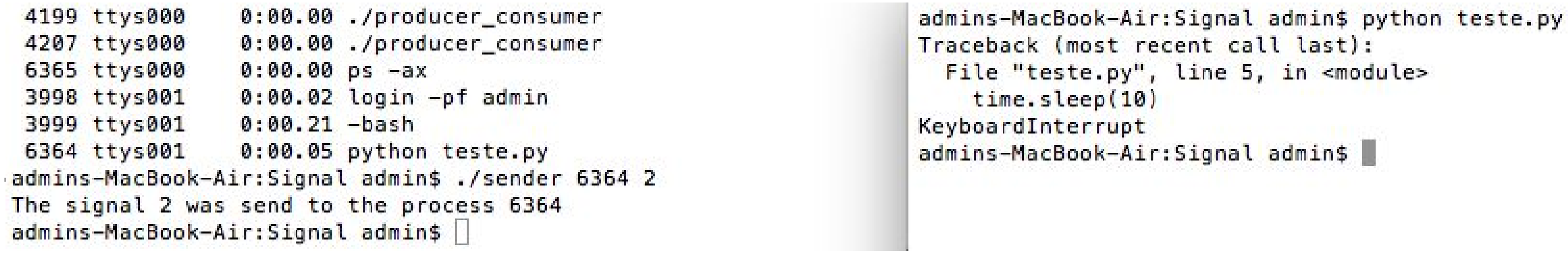
- Linguagem: C
- Bibliotecas:

Signal	Pipe	Socket

#include <signal.h>
#include <errno.h>
#include <unistd.h>
	#include <unistd.h>
	#include <time.h>
		#include <string.h>
		#include <unistd.h>
		#include <arpa/inet.h>
		#include <netinet/in.h>
		#include <sys/socket.h>
		#include <time.h>

Signal - *sender.c*

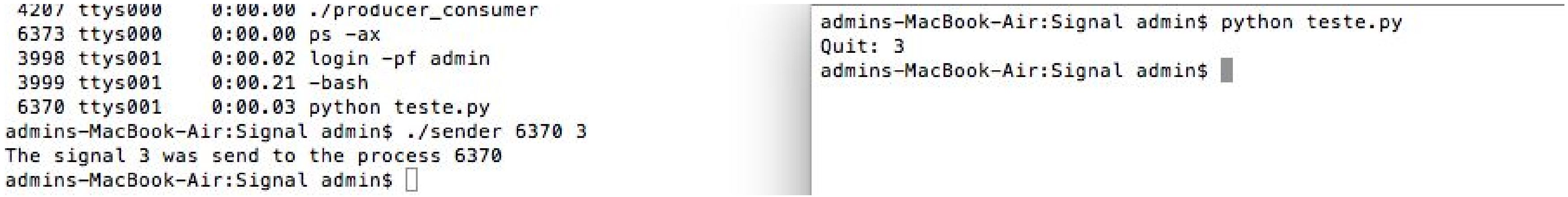
```
/* ARGUMENTS:
     argc[1] = number of the destination process
     argc[2] = signal to be send
*/


(...)


if(kill( process_number, signal_number ) != 0)
     if(errno == ESRCH)
          printf("The process %s could not be found\n", argv[1]);

     else
          printf("The signal %s was send to the process %s\n", argv[2], argv[1]);


(...)
```



```
 4199 ttys000    0:00.00 ./producer_consumer
 4207 ttys000    0:00.00 ./producer_consumer
 6365 ttys000    0:00.00 ps -ax
 3998 ttys001    0:00.02 login -pf admin
 3999 ttys001    0:00.21 -bash
 6364 ttys001    0:00.05 python teste.py
admins-MacBook-Air:Signal admin$ ./sender 6364 2
The signal 2 was send to the process 6364
admins-MacBook-Air:Signal admin$ 
```

```
admins-MacBook-Air:Signal admin$ python teste.py
Traceback (most recent call last):
  File "teste.py", line 5, in <module>
    time.sleep(10)
KeyboardInterrupt
admins-MacBook-Air:Signal admin$ 
```

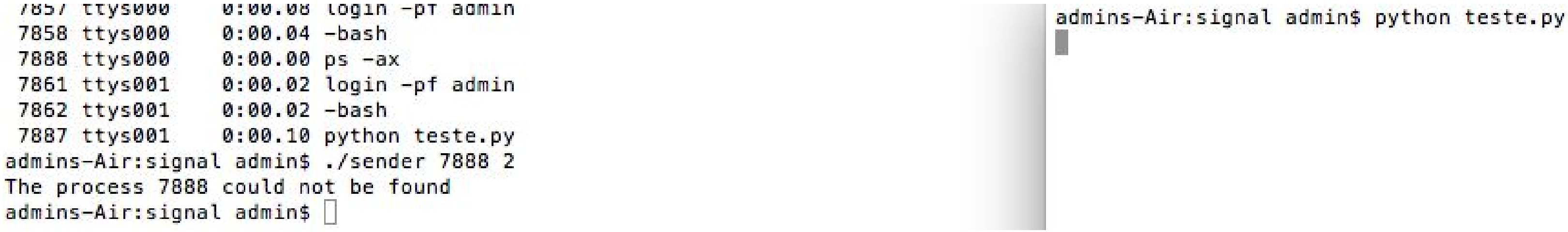
Figura 1: SIGINT (sinal de interrupção) enviado e processo interrompido corretamente



```
 4207 ttys000    0:00.00 ./producer_consumer
 6373 ttys000    0:00.00 ps -ax
 3998 ttys001    0:00.02 login -pf admin
 3999 ttys001    0:00.21 -bash
 6370 ttys001    0:00.03 python teste.py
admins-MacBook-Air:Signal admin$ ./sender 6370 3
The signal 3 was send to the process 6370
admins-MacBook-Air:Signal admin$ 
```

```
admins-MacBook-Air:Signal admin$ python teste.py
Quit: 3
admins-MacBook-Air:Signal admin$ 
```

Figura 2: SIGQUIT (sinal de término) enviado e processo terminado corretamente

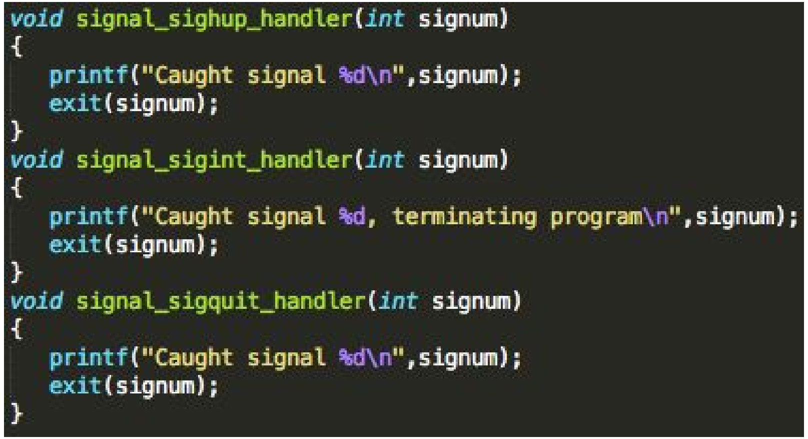


```
 7857 ttys000    0:00.08 login -pf admin
 7858 ttys000    0:00.04 -bash
 7888 ttys000    0:00.00 ps -ax
 7861 ttys001    0:00.02 login -pf admin
 7862 ttys001    0:00.02 -bash
 7887 ttys001    0:00.10 python teste.py
admins-Air:signal admin$ ./sender 7888 2
The process 7888 could not be found
admins-Air:signal admin$ 
```

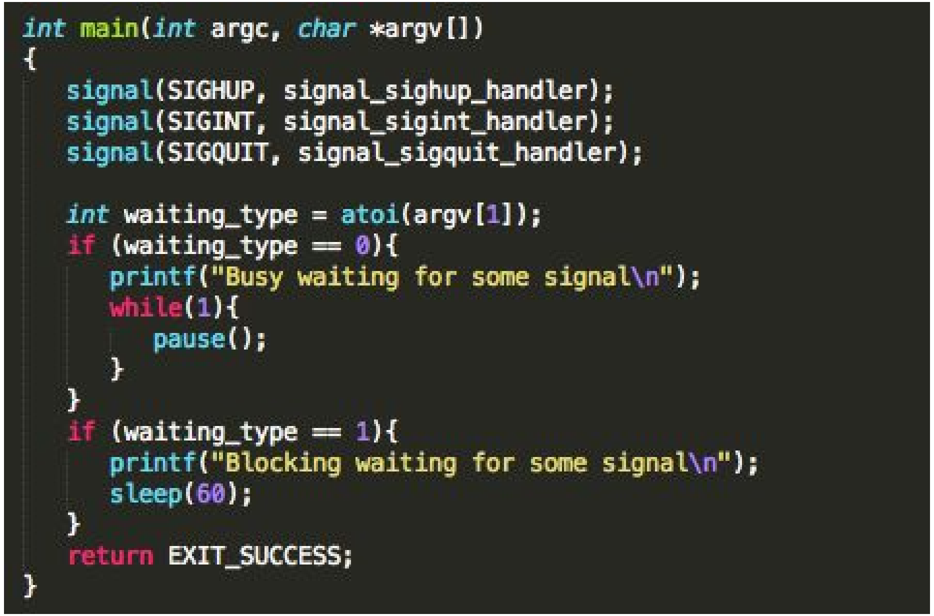
```
admins-Air:signal admin$ python teste.py
```

Figura 3: Sinal enviado para processo inexistente

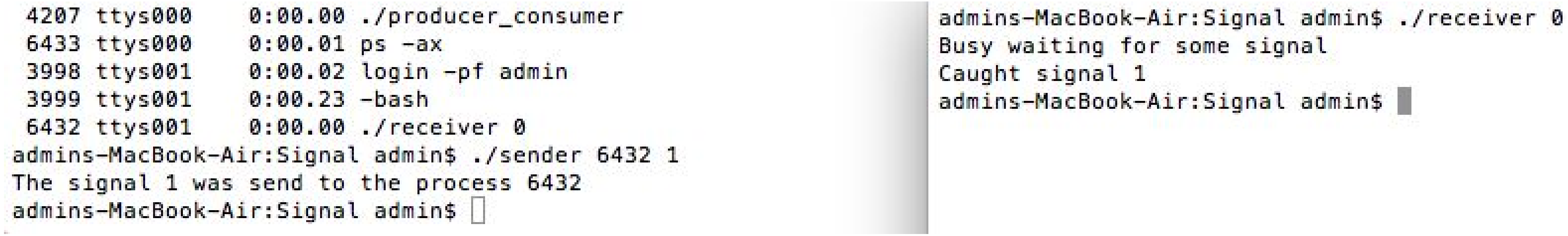
Signal - *receiver.c*



```c
void signal_sighup_handler(int signum)
{
   printf("Caught signal %d\n",signum);
   exit(signum);
}
void signal_sigint_handler(int signum)
{
   printf("Caught signal %d, terminating program\n",signum);
   exit(signum);
}
void signal_sigquit_handler(int signum)
{
   printf("Caught signal %d\n",signum);
   exit(signum);
}
```



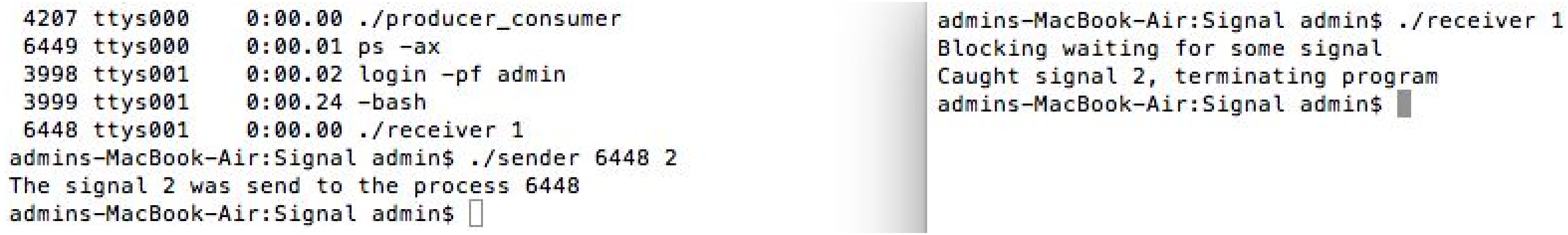
```c
int main(int argc, char *argv[])
{
   signal(SIGHUP, signal_sighup_handler);
   signal(SIGINT, signal_sigint_handler);
   signal(SIGQUIT, signal_sigquit_handler);

   int waiting_type = atoi(argv[1]);
   if (waiting_type == 0){
      printf("Busy waiting for some signal\n");
      while(1){
         pause();
      }
   }
   if (waiting_type == 1){
      printf("Blocking waiting for some signal\n");
      sleep(60);
   }
   return EXIT_SUCCESS;
}
```



```
 4207 ttys000    0:00.00 ./producer_consumer
 6433 ttys000    0:00.01 ps -ax
 3998 ttys001    0:00.02 login -pf admin
 3999 ttys001    0:00.23 -bash
 6432 ttys001    0:00.00 ./receiver 0
admins-MacBook-Air:Signal admin$ ./sender 6432 1
The signal 1 was send to the process 6432
admins-MacBook-Air:Signal admin$ 
```

```
admins-MacBook-Air:Signal admin$ ./receiver 0
Busy waiting for some signal
Caught signal 1
admins-MacBook-Air:Signal admin$ 
```

Figura 4: Envio de sinal 1 (SIGHUP) com *busy wait* como espera

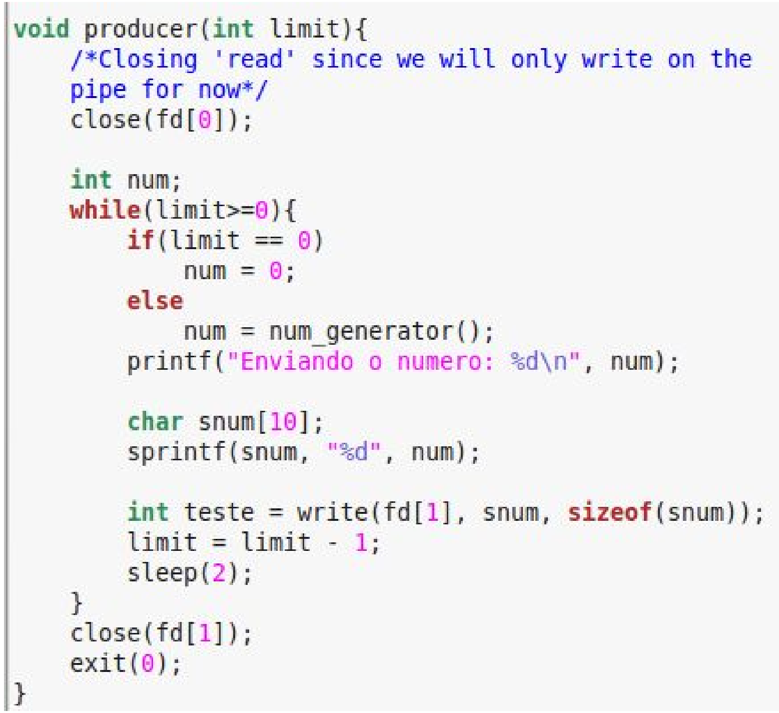


```
 4207 ttys000    0:00.00 ./producer_consumer
 6449 ttys000    0:00.01 ps -ax
 3998 ttys001    0:00.02 login -pf admin
 3999 ttys001    0:00.24 -bash
 6448 ttys001    0:00.00 ./receiver 1
admins-MacBook-Air:Signal admin$ ./sender 6448 2
The signal 2 was send to the process 6448
admins-MacBook-Air:Signal admin$ 
```

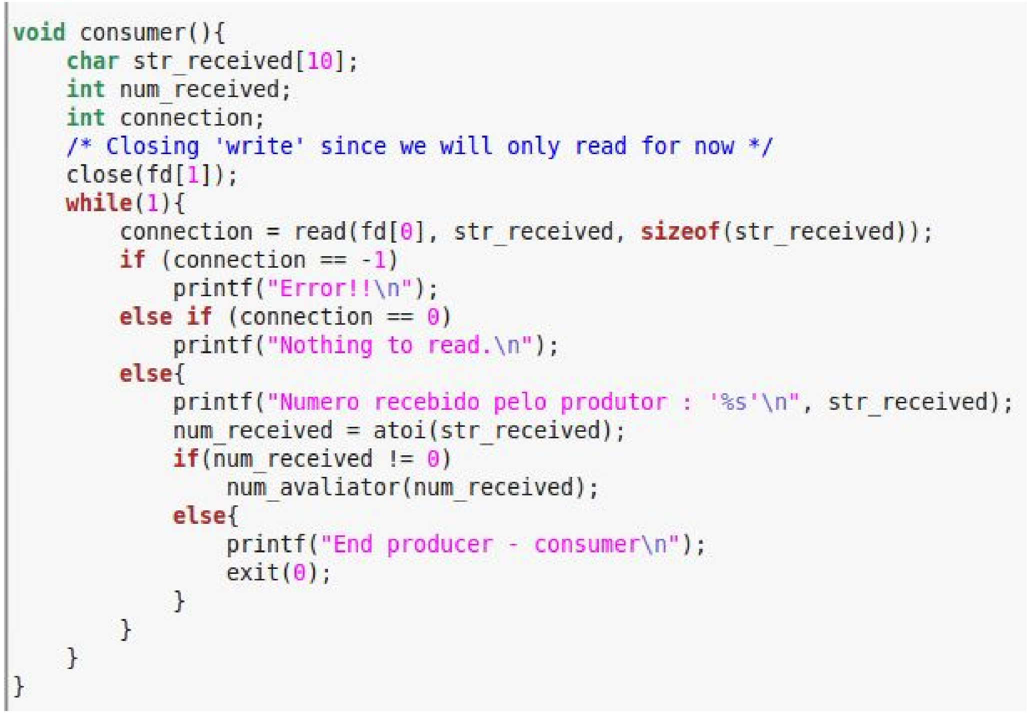
```
admins-MacBook-Air:Signal admin$ ./receiver 1
Blocking waiting for some signal
Caught signal 2, terminating program
admins-MacBook-Air:Signal admin$ 
```

Figura 5: Envio de sinal 2 (SIGINT) com *blocking wait* como espera

Pipe - *producer_consumer.c*

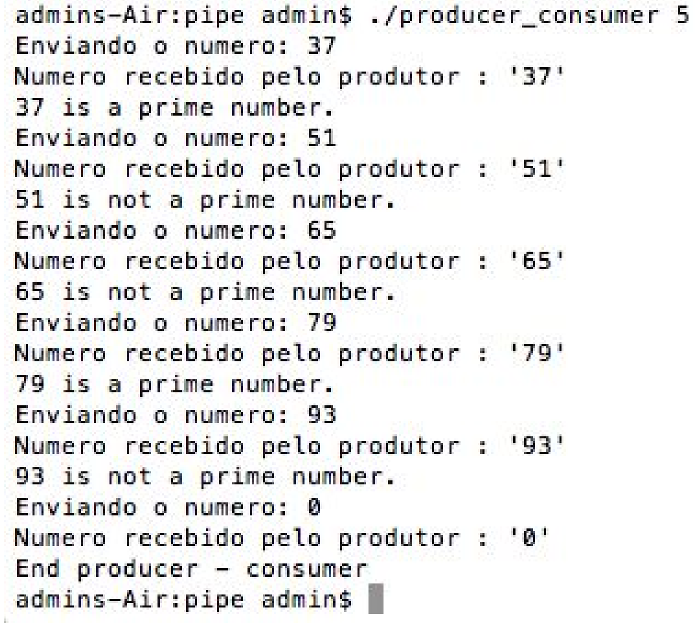


```c
void producer(int limit){
    /*Closing 'read' since we will only write on the
    pipe for now*/
    close(fd[0]);

    int num;
    while(limit>=0){
        if(limit == 0)
            num = 0;
        else
            num = num_generator();
        printf("Enviando o numero: %d\n", num);

        char snum[10];
        sprintf(snum, "%d", num);

        int teste = write(fd[1], snum, sizeof(snum));
        limit = limit - 1;
        sleep(2);
    }
    close(fd[1]);
    exit(0);
}
```



```c
void consumer(){
    char str_received[10];
    int num_received;
    int connection;
    /* Closing 'write' since we will only read for now */
    close(fd[1]);
    while(1){
        connection = read(fd[0], str_received, sizeof(str_received));
        if (connection == -1)
            printf("Error!!\n");
        else if (connection == 0)
            printf("Nothing to read.\n");
        else{
            printf("Numero recebido pelo produtor : '%s'\n", str_received);
            num_received = atoi(str_received);
            if(num_received != 0)
                num_avaliator(num_received);
            else{
                printf("End producer - consumer\n");
                exit(0);
            }
        }
    }
}
```

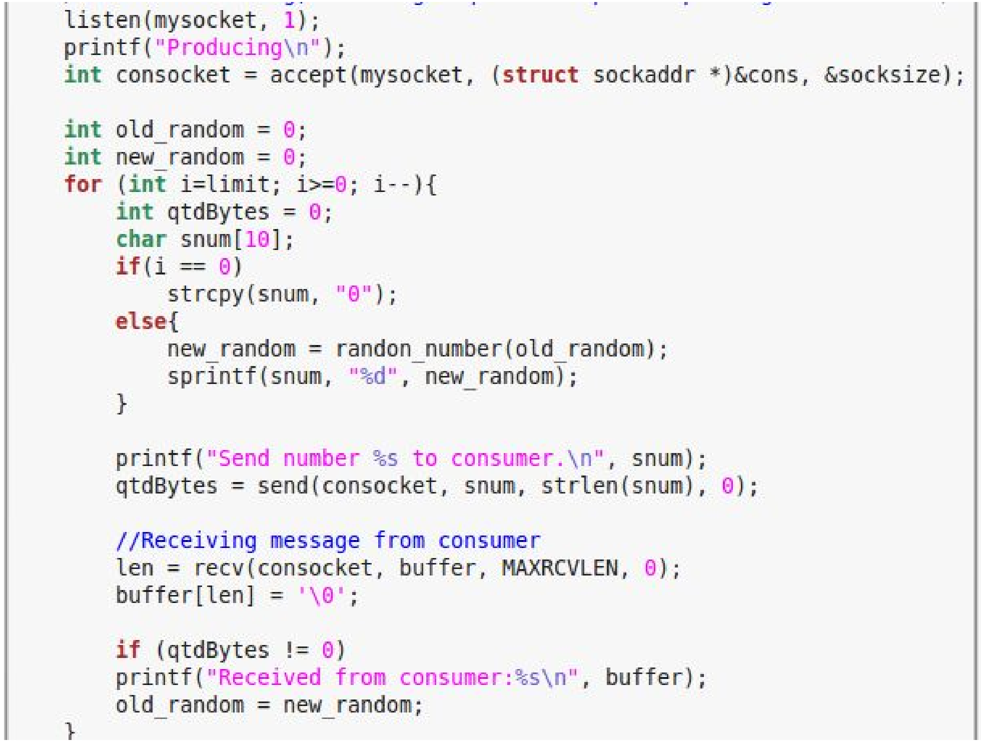
Pipe - *producer_consumer.c*



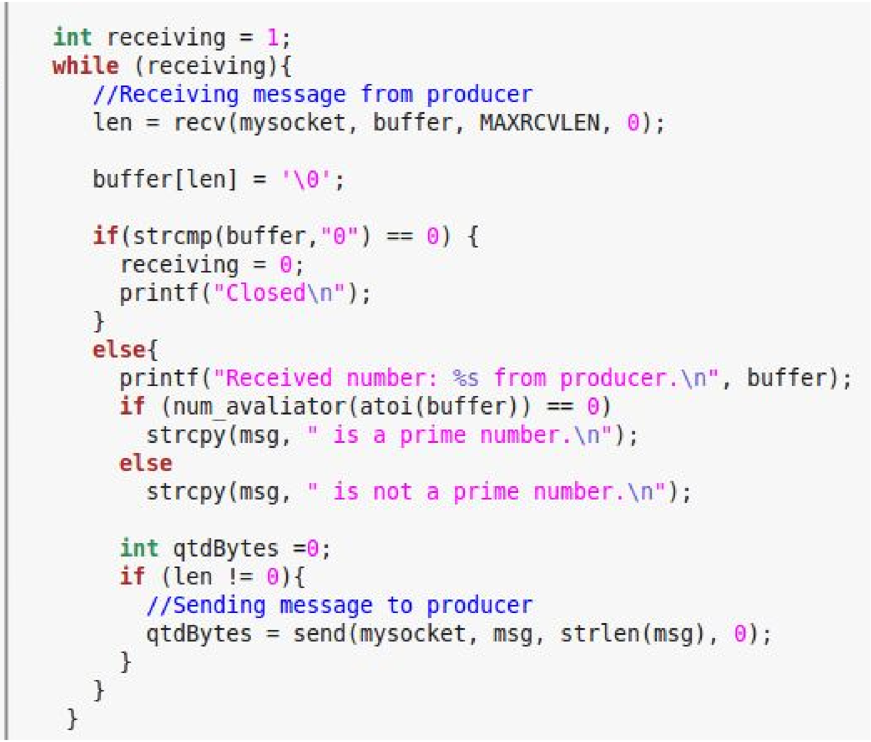
```
admins-Air:pipe admin$ ./producer_consumer 5
Enviando o numero: 37
Numero recebido pelo produtor : '37'
37 is a prime number.
Enviando o numero: 51
Numero recebido pelo produtor : '51'
51 is not a prime number.
Enviando o numero: 65
Numero recebido pelo produtor : '65'
65 is not a prime number.
Enviando o numero: 79
Numero recebido pelo produtor : '79'
79 is a prime number.
Enviando o numero: 93
Numero recebido pelo produtor : '93'
93 is not a prime number.
Enviando o numero: 0
Numero recebido pelo produtor : '0'
End producer - consumer
admins-Air:pipe admin$
```

Figura 6: Produtor - Consumidor utilizando Pipe

Socket - *producer.c e consumer.c*

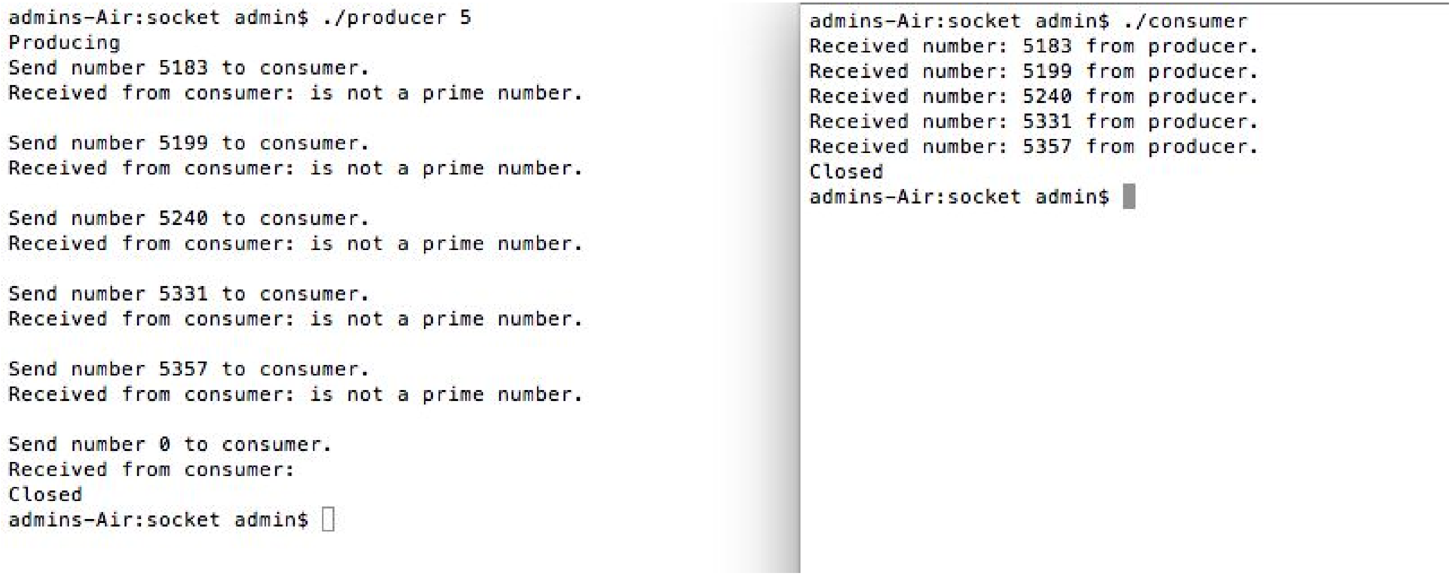


```
listen(mysocket, 1);
printf("Producing\n");
int consocket = accept(mysocket, (struct sockaddr *)&cons, &socksize);

int old_random = 0;
int new_random = 0;
for (int i=limit; i>=0; i--){
    int qtdBytes = 0;
    char snum[10];
    if(i == 0)
        strcpy(snum, "0");
    else{
        new_random = randon_number(old_random);
        sprintf(snum, "%d", new_random);
    }

    printf("Send number %s to consumer.\n", snum);
    qtdBytes = send(consocket, snum, strlen(snum), 0);

    //Receiving message from consumer
    len = recv(consocket, buffer, MAXRCVLEN, 0);
    buffer[len] = '\0';

    if (qtdBytes != 0)
    printf("Received from consumer:%s\n", buffer);
    old_random = new_random;
}
```



```
int receiving = 1;
while (receiving){
  //Receiving message from producer
  len = recv(mysocket, buffer, MAXRCVLEN, 0);

  buffer[len] = '\0';

  if(strcmp(buffer,"0") == 0) {
   receiving = 0;
   printf("Closed\n");
  }
  else{
   printf("Received number: %s from producer.\n", buffer);
   if (num_avaliator(atoi(buffer)) == 0)
    strcpy(msg, " is a prime number.\n");
   else
    strcpy(msg, " is not a prime number.\n");

   int qtdBytes =0;
   if (len != 0){
    //Sending message to producer
    qtdBytes = send(mysocket, msg, strlen(msg), 0);
   }
  }
 }
```

Socket - *producer.c e consumer.c*



```
admins-Air:socket admin$ ./producer 5
Producing
Send number 5183 to consumer.
Received from consumer: is not a prime number.

Send number 5199 to consumer.
Received from consumer: is not a prime number.

Send number 5240 to consumer.
Received from consumer: is not a prime number.

Send number 5331 to consumer.
Received from consumer: is not a prime number.

Send number 5357 to consumer.
Received from consumer: is not a prime number.

Send number 0 to consumer.
Received from consumer:
Closed
admins-Air:socket admin$ 
```

```
admins-Air:socket admin$ ./consumer
Received number: 5183 from producer.
Received number: 5199 from producer.
Received number: 5240 from producer.
Received number: 5331 from producer.
Received number: 5357 from producer.
Closed
admins-Air:socket admin$ 
```

Figura 7: Programa produtor - consumidor com Socket

Sistemas Distribuídos

COS470 2016/1

Trabalho Prático 1

Obrigada!

Débora Barbosa Pina
Patrícia de Andrade Kovaleski