Assignment 2- (FA15 BL CSCI P536 36187)

Rohit Nair – Ronair, Abhijit Karanjkar - Aykaranj

1. **Does your program output any garbage? If yes, why?**

   Answer: Yes the program randomly outputs garbage.
   Reason: The two process threads, producer and consumer are not synchronized. The execution is completely random. The processor first starts executing Producer process. Then consumer thread comes, interrupting producer thread. The context switching between two threads happens at random times which results into the printing statements randomly from two processes on command prompt. Moreover printf() itself is not synchronized due to which sometimes before completing the print statement of one process the print statement of other process gets executed. For example, if the Producer wants to print “Produced : 5” and consumer wants to print “Consumed : 4” then according to the explanation given above sometimes it may output as Produc4ed : 5sumed.
   So to conclude that, the garbage values we are seeing are not actually garbage values, but those are the values randomly printed by two non-synchronized threads.

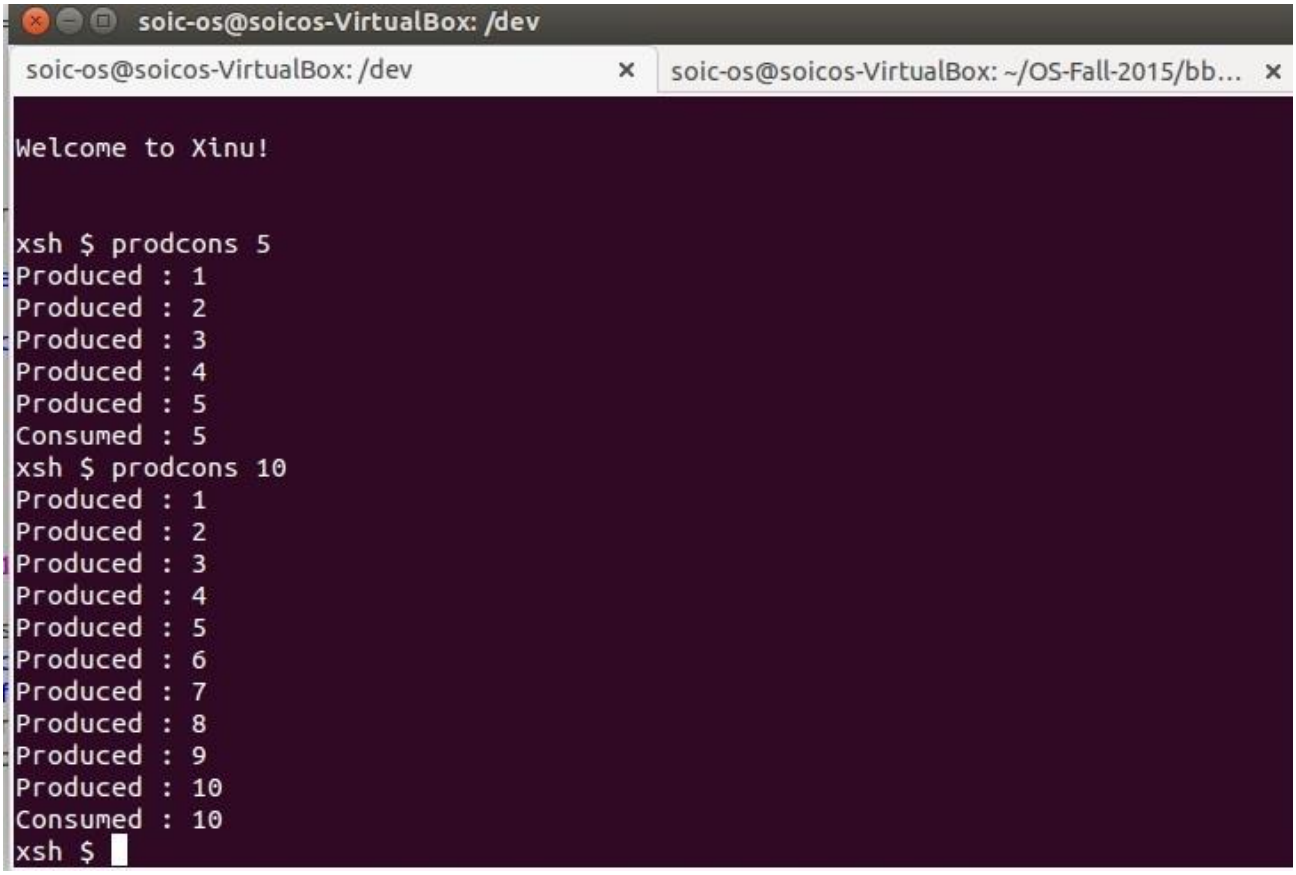
2. **Are all produced values getting consumed? Check your program for small count 20.**

   Answer: Not all values are getting consumed even for small program. We have tried putting count values as 5, 10, 15 and 20. Most of the times only last value gets consumed.
   Please see the screenshots below:
   For count = 5:

Assignment 2- (FA15 BL CSCI P536 36187)

Rohit Nair – Ronair, Abhijit Karanjkar - Aykaranj

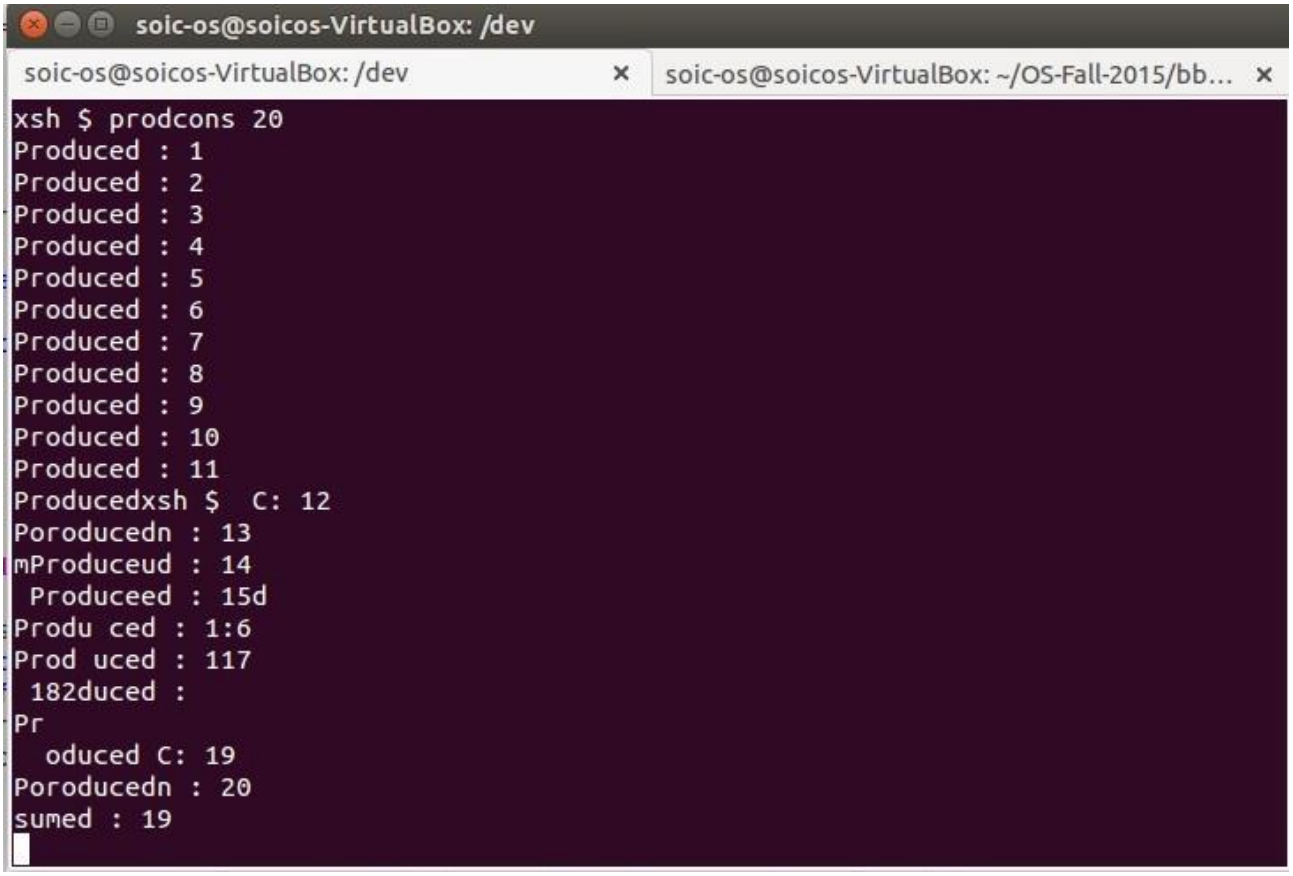
For count = 10:



For count = 10:

Assignment 2- (FA15 BL CSCI P536 36187)

Rohit Nair – Ronair, Abhijit Karanjkar - Aykaranj



3. Program Functions :

1. **isNumber(const char *val):** To check if a character is number

```
/*
Check if character is number
*/
int isNumber(const char *val)   {

     while(*val != '\0')   {
     if(*val < '0' || *val > '9')   {
        return 0;
     }
     val++;
  }

  return 1;
}
```

2. **xsh_prodcons(int nargs, char *args[]):** Prodcons shell command

```
shellcmd xsh_prodcons(int nargs, char *args[])
{
    //Argument verifications and validations

    int count = 2000;           //local varible to hold count
  n = 0;


  if (nargs == 2) {

     if(strncmp(args[1], "--help", 7) == 0)   {
        printf("\nThis command executes producer & consumer!\n");
  }else   {
     if(isNumber(args[1]) == 1)   {

        count = atoi(args[1]);
        //create the process producer and consumer and put them in ready queue.
         //Look at the definations of function create and resume in exinu/system folder for
reference.
            resume( create(producer, 1024, 20, "producer", 1, count) );
            resume( create(consumer, 1024, 20, "consumer", 1, count) );

     }else   {
```

```
            printf("\nIncorrect value entered! Please inter the numeric value.\n");
        }


  }}else if(nargs == 1)   {

        resume( create(producer, 1024, 20, "producer", 1, count) );
            resume( create(consumer, 1024, 20, "consumer", 1, count) );
  } else   {

     printf("\nMore than one parameters are not allowed!\n");
  }



}
```

3. **producer(int count):** Produces the value of n.

```
void producer(int count)
{

        int32 i;

        for(i = 1; i <= count; i++)  {

                n++;
                printf("Produced : %d\n", n);

        }
}
```

4. **consumer(int count):** Consumes the value of n.

```
void consumer(int count)
{
```

```
        int32 i;
        for(i = 1; i <= count; i++)  {


                printf("Consumed : %d\n", n);

                if(n == count)          {
                        break;
        }

        }
}
```

4. Teamwork

Abhijit Karanjkar(aykaranj)	**Rohit Nair(ronair)**
Implemented xsh_prodcons	Created Method skeleton and initial setup
Implemented Producer	Implemented Consumer
Implemented Field validation	Tested output for different count values
Discussed with Rohit and prepared answer for question 1	Discussed with Abhijit and prepared answer for question 2