

## Assignment 2 Report

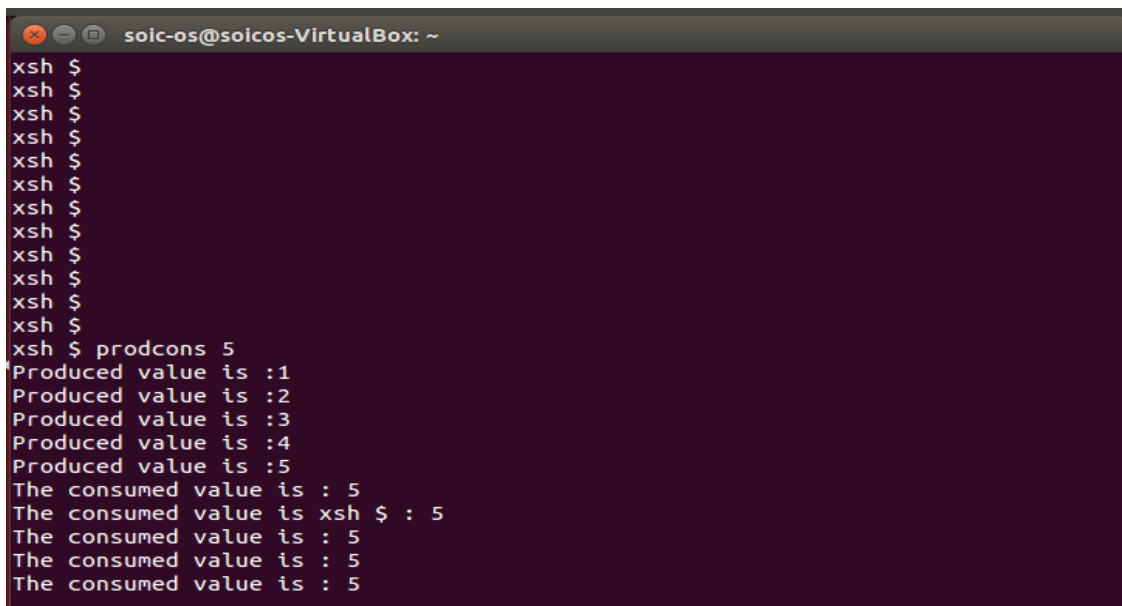
---

### Does your program output any garbage? If yes, why?

Though both the producer and consumer processes run concurrently they both take different processor time to execute a machine instruction per iteration. Consumer takes more time than the producer as it needs to format and output the values produced by producer process. While performing output operation consumer fills all the available output buffers and waits for the output device to send all the characters to console. While the consumer waits the producer runs. In our program the producer runs faster than the consumer process and both the processes are using a common buffer (producer to write in the buffer and consumer to read and print from the buffer). And we are using `printf()` to print the values in the consumer code. The output we get is sometimes overlapping with the previous output which is caused because of the fact that the buffer is not getting flushed properly before the context switch happens between the two processes and hence the overlapping of values occur.

### Are all the produced values getting consumed? Check your program for a small count like 20.

Not all the produced values are getting consumed because of the fact that producer executes only a few machine instructions per iteration. The producer runs through its entire loop and exits in the short time it takes the console device to send a line of characters. When the consumer resumes execution again, it finds that `n` has increased in value.



```
soic-os@soicos-VirtualBox: ~
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $
xsh $ prodcons 5
Produced value is :1
Produced value is :2
Produced value is :3
Produced value is :4
Produced value is :5
The consumed value is : 5
The consumed value is xsh $ : 5
The consumed value is : 5
The consumed value is : 5
The consumed value is : 5
```

## Assignment 2 Report

---

### **producer.c**

```
#include <prodcons.h>
void producer(int count)
{
    int i;
    for(i=1; i<=count; i++)
    {
        n=i;
        printf("Produced value is :%d\n", n);
    }
}
```

### **consumer.c**

```
#include<prodcons.h>
void consumer(int count)
{
    int i;
    for(i=1; i<=count; i++)
        printf("The consumed value is : %d \n",n);
}
```

### **xsh\_prodcons.c**

```
#include <prodcons.h>
int n;
shellcmd xsh_prodcons(int nargs, char *args[])
{
    int count;
    if(nargs>2){
        printf("Too many arguments.\n");
        return 0;
    }
    if(nargs==2){
        count=atoi(args[1]);
    }
    else
        count=2000;
    resume(create(producer, 1024, 20, "producer", 1, count));
    resume(create(consumer, 1024, 20, "consumer", 1, count));
    return 0;
}
```

## Assignment 2 Report

---

### **prodcons.h**

```
#include <xinu.h>
```

```
extern int n;
```

```
void producer(int count);  
void consumer(int count);
```

### **NOTE:**

**xsh\_prodcons.c: Rudrani, Kushal**

**producer.c: Rudrani**

**consumer.c: Kushal**

**Report: Rudrani, Kushal**