P536: ADVANCED OPERATING SYSTEMS

ASSIGNMENT 2: REPORT

TEAM: ROHIT PATIL, SAMEEKSHA VAITY

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

Answer: Yes. Our program does produce garbage values.

The program outputs garbage values because in this program the same global variable n is being accessed by the consumer and the producer at the same time. This confuses the operating system as to which operation is to be performed whether it should first increment (produce) the value or the value should be passed for consumption (display). The concurrent execution of two processes on the same variable causes the program to produce garbage values. The printf command is not a synchronized command and both the consumer producer functions use the same output buffer and hence it produces garbage values.

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

Answer: No. All the produced values are not getting consumed.

The reason being that even though the two processes are running concurrently the time required by each of the processes to complete the operation is different. Here the consumer is performing the action of outputting or filling the output buffer at a higher rate than at which the output device (console) is able to print. So then the consumer has to wait till the console has done printing the character but till then the producer is still running and producing the values and it takes a shorter time to complete its operation than the time taken by the console to print. So by the time console is done printing and consumer resumes back the producer is already done producing and the consumer just gets the last value to consume and hence consumes just the last value produced.

FUNCTIONS USED IN THE PROJECT:

```
1.  void produce(int count)
    {
            for(int i = 1;i<count;i++)
            {
             n=i;
             printf("Produced value : %d\n,n");

            }

    }
2.  void consumer(int count)
    {
            for(int i = 1;i<count;i++)
            {
            printf("Consumed value : %d\n,n");
            }
    }
3.  int isNumeric(const char *str)
      {
       while(*str != '\0')
         {
           if(*str < '0' || *str > '9')
           return 0;
           str++;
         }
      return 1;
       }
```

TASKS OF THE GROUP MEMBERS:

Rohit – Developed the producer class logic and created xsh_prodcons.h, prodcons.h header files which were to be included in the individual producer and consumer classes

Sameeksha – Developed the consumer class logic and implemented the input arguments to be assigned to count functionality and updated the shell.c file to include the prodcons command to be executed.