# CSE 344 SYSTEMS PROGRAMMING

**HW 4 REPORT**

**151044004**

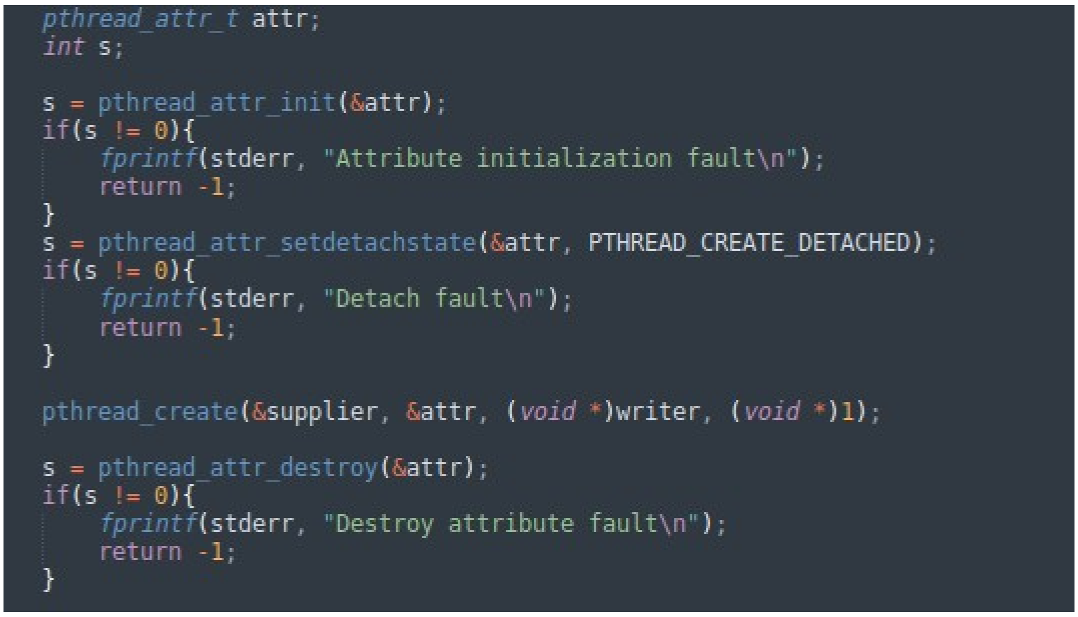
**MURAT BEYAZ**

## Problem Definition

There is one supplier thread and multiple consumer threads. The supplier brings materials, one by one. And the consumers consume them, two by two. Each actor will be modeled by its own thread.

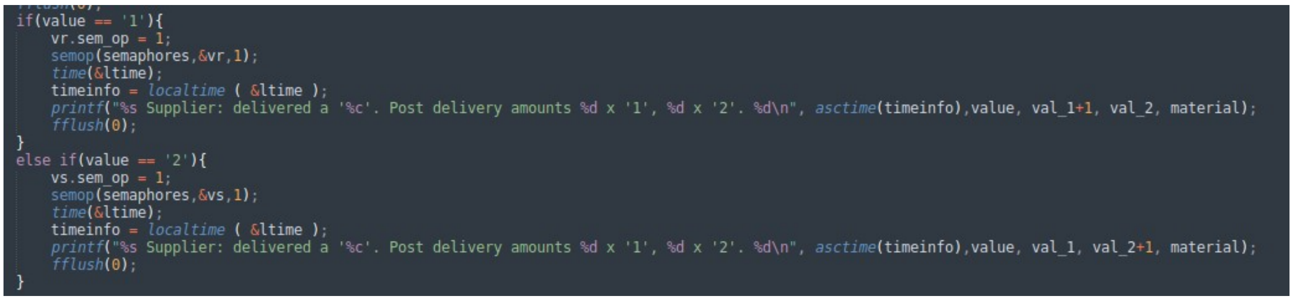
Synchronization between these threads will be obtained via system V semaphores. Producer thread will be a detached thread and consumer threads will be non detached threads.

**Supplier:** Supplier thread will read the input file's contents one character at a time and output a message concerning its activity. For every character read from the file, it will post a semaphore representing it's amount. It will terminate once it reaches the end of the file. Supplier is a detached thread. Code section that makes the thread detached is shown below:



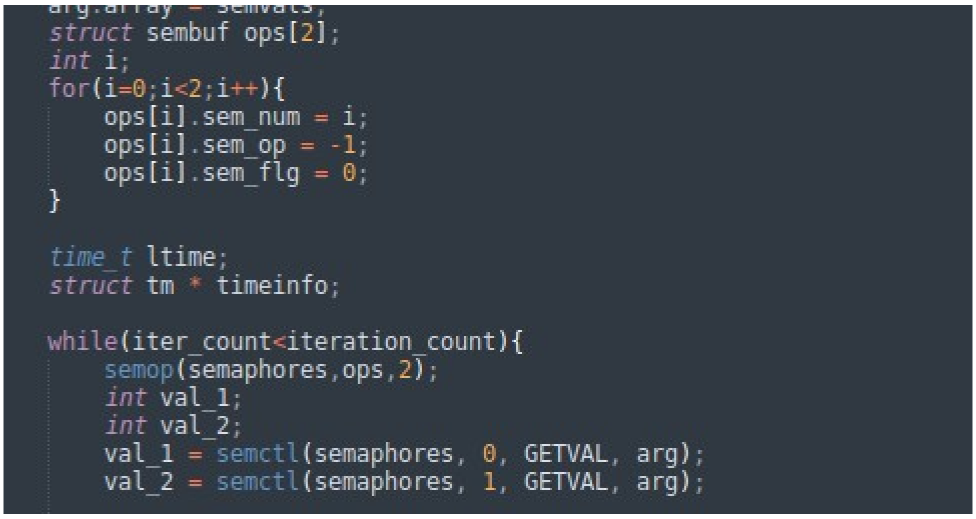
```c
pthread_attr_t attr;
int s;

s = pthread_attr_init(&attr);
if(s != 0){
    fprintf(stderr, "Attribute initialization fault\n");
    return -1;
}
s = pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_DETACHED);
if(s != 0){
    fprintf(stderr, "Detach fault\n");
    return -1;
}

pthread_create(&supplier, &attr, (void *)writer, (void *)1);

s = pthread_attr_destroy(&attr);
if(s != 0){
    fprintf(stderr, "Destroy attribute fault\n");
    return -1;
}
```

This code section is taken from the course material.
When the supplier reads from the file it posts the corresponding semaphore to the material. Shown below:



```c
if(value == '1'){
    vr.sem_op = 1;
    semop(semaphores,&vr,1);
    time(&ltime);
    timeinfo = localtime ( &ltime );
    printf("%s Supplier: delivered a '%c'. Post delivery amounts %d x '1', %d x '2'. %d\n", asctime(timeinfo),value, val_1+1, val_2, material);
    fflush(0);
}
else if(value == '2'){
    vs.sem_op = 1;
    semop(semaphores,&vs,1);
    time(&ltime);
    timeinfo = localtime ( &ltime );
    printf("%s Supplier: delivered a '%c'. Post delivery amounts %d x '1', %d x '2'. %d\n", asctime(timeinfo),value, val_1, val_2+1, material);
    fflush(0);
}
```

**Consumers:** Each consumer will have its' own thread. It's task is to loop N times. At each iteration it will take one '1' and one '2' by reducing the corresponding semaphore values. Consumers can take the materials only if both occur at the same time. So that the decrementation of the semaphore must be atomic to avoid synchronization problems. Consumer threads waiting until both 1 and 2 occur and one of them wakes up when it does. This synchronization code segment is shown below:



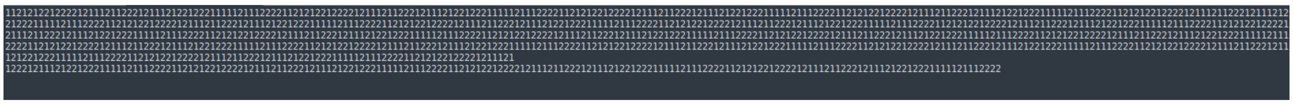
```c
arg.array = semvals;
struct sembuf ops[2];
int i;
for(i=0;i<2;i++){
    ops[i].sem_num = i;
    ops[i].sem_op = -1;
    ops[i].sem_flg = 0;
}

time_t ltime;
struct tm * timeinfo;

while(iter_count<iteration_count){
    semop(semaphores,ops,2);
    int val_1;
    int val_2;
    val_1 = semctl(semaphores, 0, GETVAL, arg);
    val_2 = semctl(semaphores, 1, GETVAL, arg);
```

As you can see the code segment above I've created a 2 (one for each semaphore) sized sembuf struct. And at the top of the loop with semop function I am sending the sembuf struct with 'wait' operation. This method provides atomicity so that threads does not decrement semaphores until both are bigger than 1. Since one of the semaphore is 0(means blocking) the thread is blocked.

## TEST

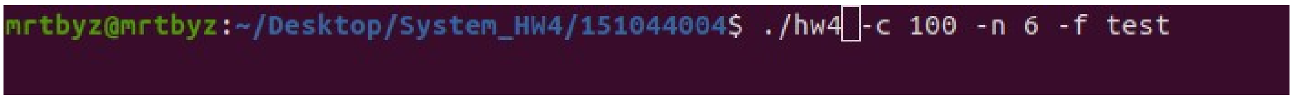
### Test 1:

First test file is shown below. There are 1200 characters 600 is '1' and 600 is '2'.



```
1121212212222121112112221211121221222111112111222211212122122221211121122212111212212221111121112222112121221222212111211222121112122122211111211122221121212212222121112112221211121221222111112111222211212122122221211121122212111212
2122211111211122221121212212222121112112221211121221222111112111222211212122122221211121122212111212212221111121112222112121221222212111211222121112122122211111211122221121212212222121112112221211121221222111112111222211212122122221
2111211222121112122122211111211122221121212212222121112112221211121221222111112111222211212122122221211121122212111212212221111121112222112121221222212111211222121112122122211111211122221121212212222121112112221211121221222111112111
2222112121221222212111211222121112122122211111211122221121212212222121112112221211121221222111112111222211212122122221211121122212111212212221111121112222112121221222212111211222121112122122211111211122221121212212222121112112221211
12122122211111211122221121212212222121112112221211121221222111112111222211212122122221211121
122212111212212221111121112222112121221222212111211222121112122122211111211122221121212212222121112112221211121221222111112111222211212122122221211121122212111212212221111121112222
```

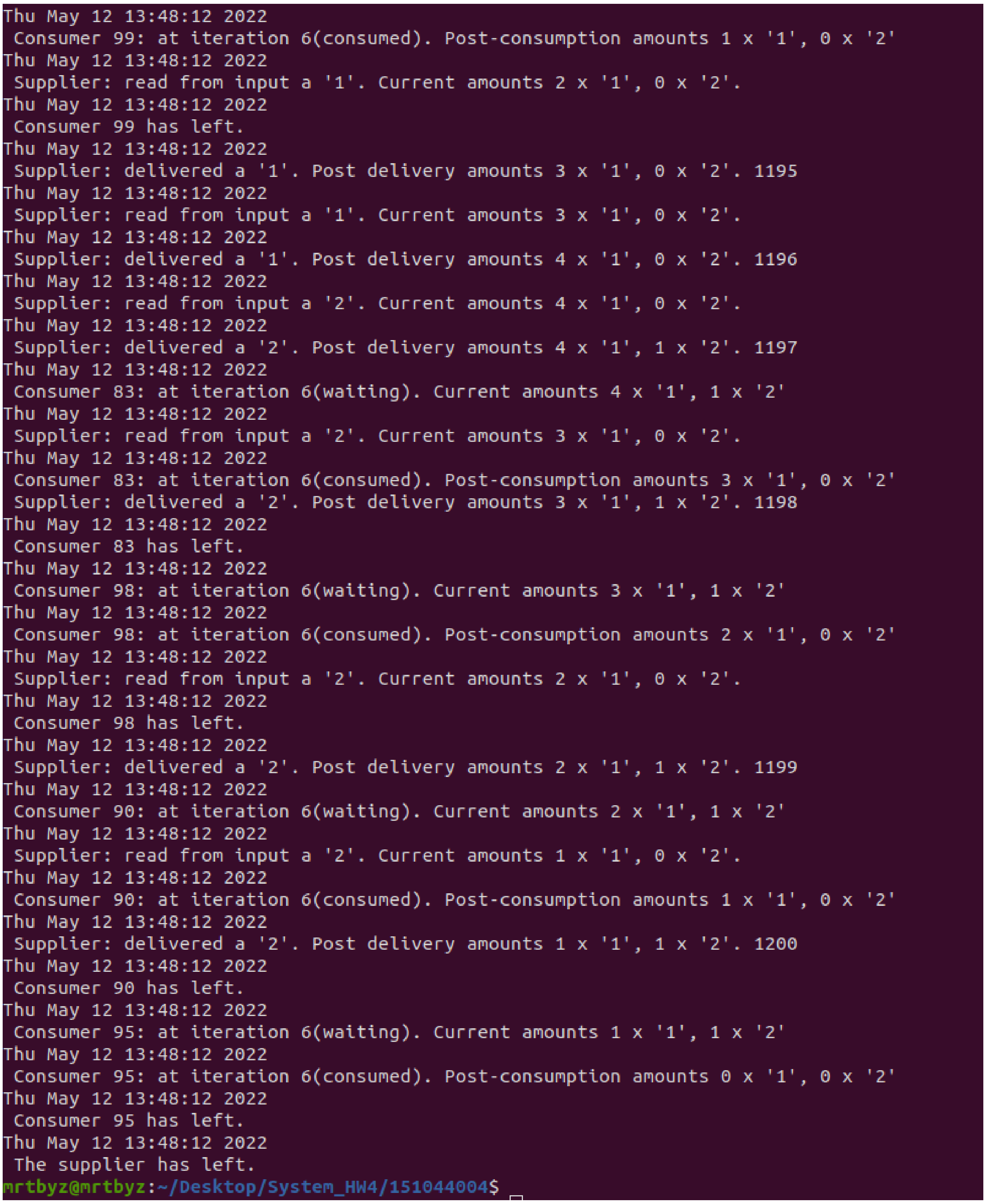
Command to run the program is shown below:



```
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ ./hw4 -c 100 -n 6 -f test
```

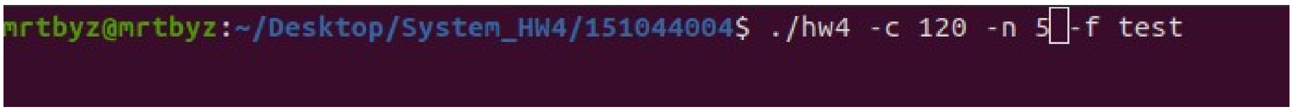
There are 100 consumers and each will consume 6 times. Equals to 1200 character 600 '1's and 600 '2's.

Result is shown below:



```
Thu May 12 13:48:12 2022
 Consumer 99: at iteration 6(consumed). Post-consumption amounts 1 x '1', 0 x '2'
Thu May 12 13:48:12 2022
 Supplier: read from input a '1'. Current amounts 2 x '1', 0 x '2'.
Thu May 12 13:48:12 2022
 Consumer 99 has left.
Thu May 12 13:48:12 2022
 Supplier: delivered a '1'. Post delivery amounts 3 x '1', 0 x '2'. 1195
Thu May 12 13:48:12 2022
 Supplier: read from input a '1'. Current amounts 3 x '1', 0 x '2'.
Thu May 12 13:48:12 2022
 Supplier: delivered a '1'. Post delivery amounts 4 x '1', 0 x '2'. 1196
Thu May 12 13:48:12 2022
 Supplier: read from input a '2'. Current amounts 4 x '1', 0 x '2'.
Thu May 12 13:48:12 2022
 Supplier: delivered a '2'. Post delivery amounts 4 x '1', 1 x '2'. 1197
Thu May 12 13:48:12 2022
 Consumer 83: at iteration 6(waiting). Current amounts 4 x '1', 1 x '2'
Thu May 12 13:48:12 2022
 Supplier: read from input a '2'. Current amounts 3 x '1', 0 x '2'.
Thu May 12 13:48:12 2022
 Consumer 83: at iteration 6(consumed). Post-consumption amounts 3 x '1', 0 x '2'
 Supplier: delivered a '2'. Post delivery amounts 3 x '1', 1 x '2'. 1198
Thu May 12 13:48:12 2022
 Consumer 83 has left.
Thu May 12 13:48:12 2022
 Consumer 98: at iteration 6(waiting). Current amounts 3 x '1', 1 x '2'
Thu May 12 13:48:12 2022
 Consumer 98: at iteration 6(consumed). Post-consumption amounts 2 x '1', 0 x '2'
Thu May 12 13:48:12 2022
 Supplier: read from input a '2'. Current amounts 2 x '1', 0 x '2'.
Thu May 12 13:48:12 2022
 Consumer 98 has left.
Thu May 12 13:48:12 2022
 Supplier: delivered a '2'. Post delivery amounts 2 x '1', 1 x '2'. 1199
Thu May 12 13:48:12 2022
 Consumer 90: at iteration 6(waiting). Current amounts 2 x '1', 1 x '2'
Thu May 12 13:48:12 2022
 Supplier: read from input a '2'. Current amounts 1 x '1', 0 x '2'.
Thu May 12 13:48:12 2022
 Consumer 90: at iteration 6(consumed). Post-consumption amounts 1 x '1', 0 x '2'
Thu May 12 13:48:12 2022
 Supplier: delivered a '2'. Post delivery amounts 1 x '1', 1 x '2'. 1200
Thu May 12 13:48:12 2022
 Consumer 90 has left.
Thu May 12 13:48:12 2022
 Consumer 95: at iteration 6(waiting). Current amounts 1 x '1', 1 x '2'
Thu May 12 13:48:12 2022
 Consumer 95: at iteration 6(consumed). Post-consumption amounts 0 x '1', 0 x '2'
Thu May 12 13:48:12 2022
 Consumer 95 has left.
Thu May 12 13:48:12 2022
 The supplier has left.
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$
```

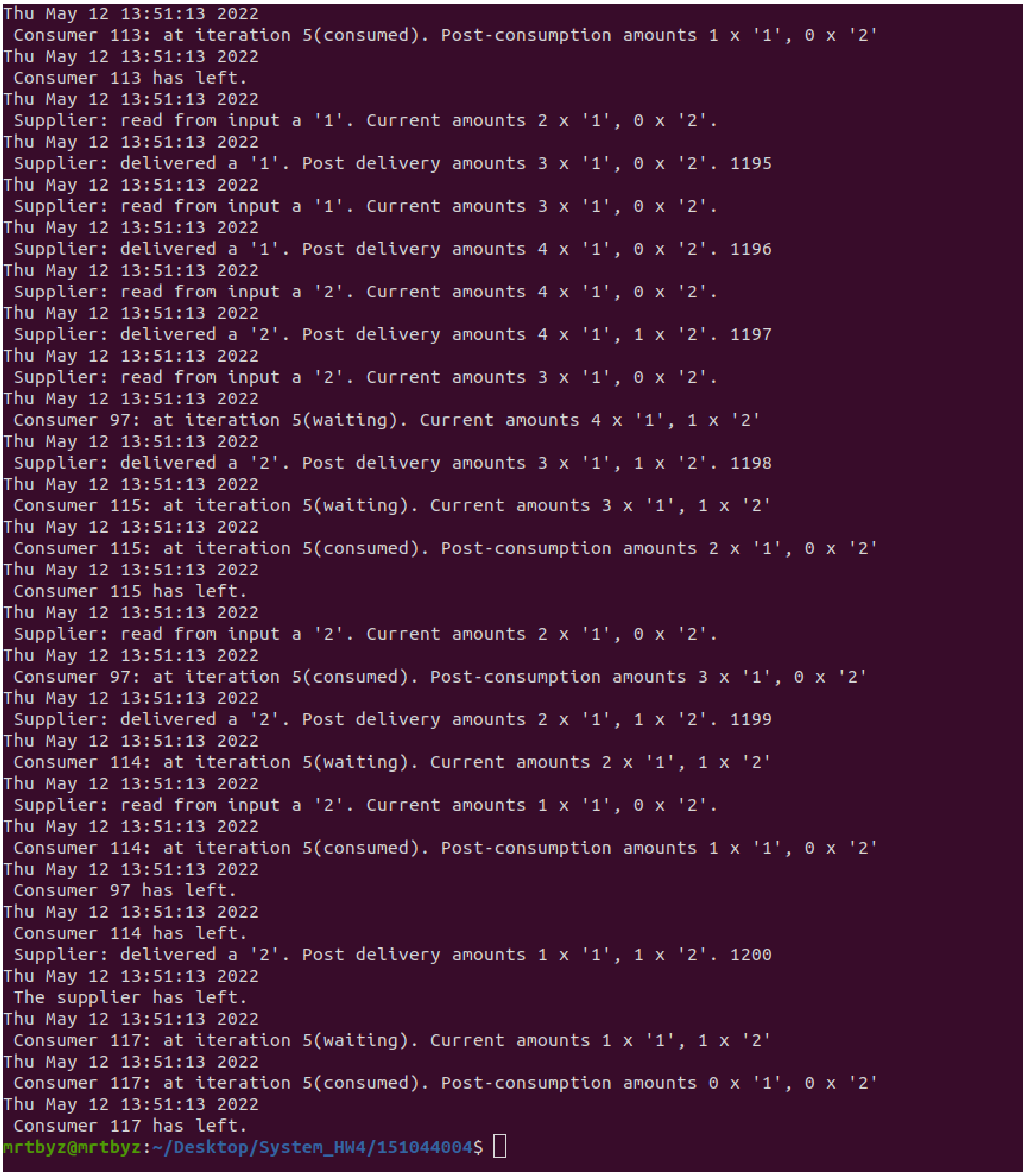
Same file is tested with a different command shown below:



```
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ ./hw4 -c 120 -n 5 -f test
```

There are 120 consumers and each will consume 5 times. Equals to 1200 characters.

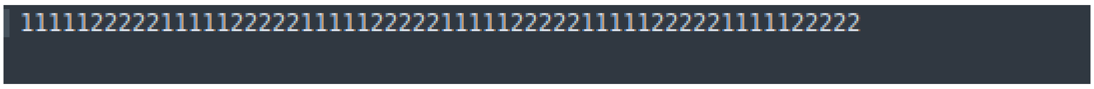
Result is shown below:



```
Thu May 12 13:51:13 2022
 Consumer 113: at iteration 5(consumed). Post-consumption amounts 1 x '1', 0 x '2'
Thu May 12 13:51:13 2022
 Consumer 113 has left.
Thu May 12 13:51:13 2022
 Supplier: read from input a '1'. Current amounts 2 x '1', 0 x '2'.
Thu May 12 13:51:13 2022
 Supplier: delivered a '1'. Post delivery amounts 3 x '1', 0 x '2'. 1195
Thu May 12 13:51:13 2022
 Supplier: read from input a '1'. Current amounts 3 x '1', 0 x '2'.
Thu May 12 13:51:13 2022
 Supplier: delivered a '1'. Post delivery amounts 4 x '1', 0 x '2'. 1196
Thu May 12 13:51:13 2022
 Supplier: read from input a '2'. Current amounts 4 x '1', 0 x '2'.
Thu May 12 13:51:13 2022
 Supplier: delivered a '2'. Post delivery amounts 4 x '1', 1 x '2'. 1197
Thu May 12 13:51:13 2022
 Supplier: read from input a '2'. Current amounts 3 x '1', 0 x '2'.
Thu May 12 13:51:13 2022
 Consumer 97: at iteration 5(waiting). Current amounts 4 x '1', 1 x '2'
Thu May 12 13:51:13 2022
 Supplier: delivered a '2'. Post delivery amounts 3 x '1', 1 x '2'. 1198
Thu May 12 13:51:13 2022
 Consumer 115: at iteration 5(waiting). Current amounts 3 x '1', 1 x '2'
Thu May 12 13:51:13 2022
 Consumer 115: at iteration 5(consumed). Post-consumption amounts 2 x '1', 0 x '2'
Thu May 12 13:51:13 2022
 Consumer 115 has left.
Thu May 12 13:51:13 2022
 Supplier: read from input a '2'. Current amounts 2 x '1', 0 x '2'.
Thu May 12 13:51:13 2022
 Consumer 97: at iteration 5(consumed). Post-consumption amounts 3 x '1', 0 x '2'
Thu May 12 13:51:13 2022
 Supplier: delivered a '2'. Post delivery amounts 2 x '1', 1 x '2'. 1199
Thu May 12 13:51:13 2022
 Consumer 114: at iteration 5(waiting). Current amounts 2 x '1', 1 x '2'
Thu May 12 13:51:13 2022
 Supplier: read from input a '2'. Current amounts 1 x '1', 0 x '2'.
Thu May 12 13:51:13 2022
 Consumer 114: at iteration 5(consumed). Post-consumption amounts 1 x '1', 0 x '2'
Thu May 12 13:51:13 2022
 Consumer 97 has left.
Thu May 12 13:51:13 2022
 Consumer 114 has left.
 Supplier: delivered a '2'. Post delivery amounts 1 x '1', 1 x '2'. 1200
Thu May 12 13:51:13 2022
 The supplier has left.
Thu May 12 13:51:13 2022
 Consumer 117: at iteration 5(waiting). Current amounts 1 x '1', 1 x '2'
Thu May 12 13:51:13 2022
 Consumer 117: at iteration 5(consumed). Post-consumption amounts 0 x '1', 0 x '2'
Thu May 12 13:51:13 2022
 Consumer 117 has left.
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$
```

**Test 2:**

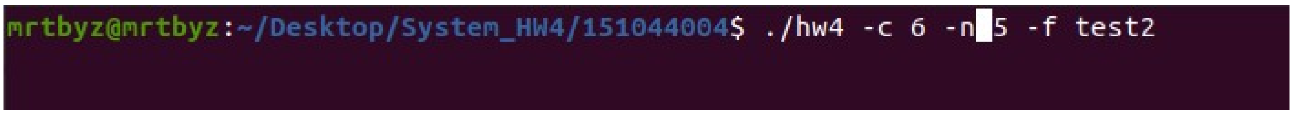
Test file is shown below:



```
111112222211111222221111122222111112222211111222221111122222
```

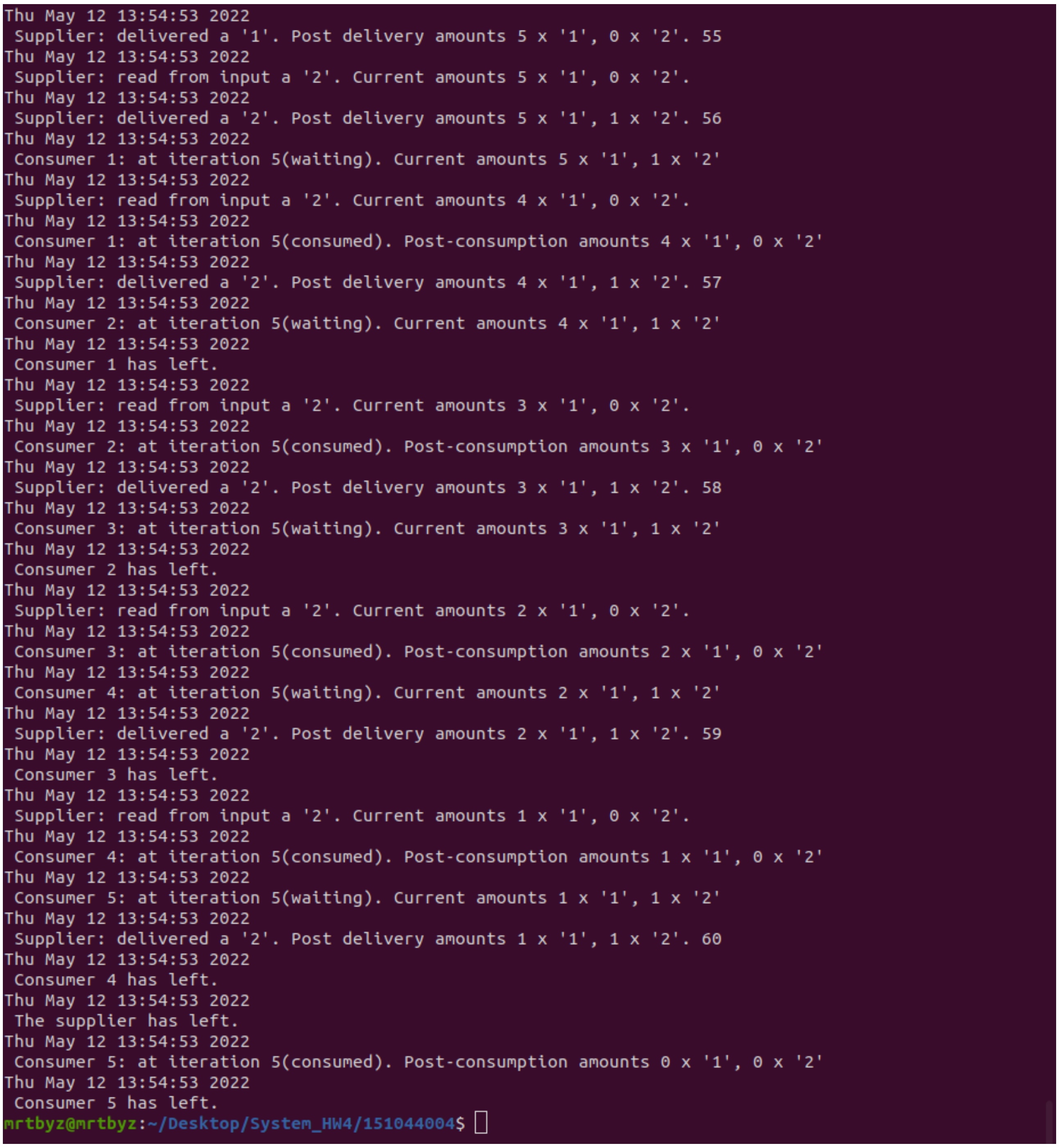
There are 60 characters 30 ‘1’s and 30 ‘2’s.

Command to run the program is shown below:



```
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ ./hw4 -c 6 -n 5 -f test2
```

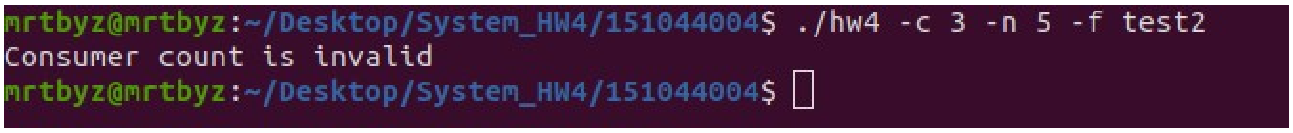
Result is shown below:



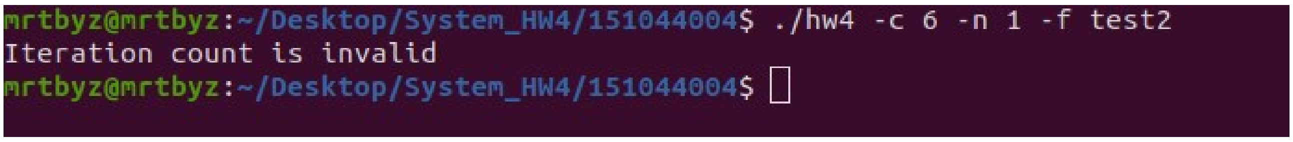
```
Thu May 12 13:54:53 2022
 Supplier: delivered a '1'. Post delivery amounts 5 x '1', 0 x '2'. 55
Thu May 12 13:54:53 2022
 Supplier: read from input a '2'. Current amounts 5 x '1', 0 x '2'.
Thu May 12 13:54:53 2022
 Supplier: delivered a '2'. Post delivery amounts 5 x '1', 1 x '2'. 56
Thu May 12 13:54:53 2022
 Consumer 1: at iteration 5(waiting). Current amounts 5 x '1', 1 x '2'
Thu May 12 13:54:53 2022
 Supplier: read from input a '2'. Current amounts 4 x '1', 0 x '2'.
Thu May 12 13:54:53 2022
 Consumer 1: at iteration 5(consumed). Post-consumption amounts 4 x '1', 0 x '2'
Thu May 12 13:54:53 2022
 Supplier: delivered a '2'. Post delivery amounts 4 x '1', 1 x '2'. 57
Thu May 12 13:54:53 2022
 Consumer 2: at iteration 5(waiting). Current amounts 4 x '1', 1 x '2'
Thu May 12 13:54:53 2022
 Consumer 1 has left.
Thu May 12 13:54:53 2022
 Supplier: read from input a '2'. Current amounts 3 x '1', 0 x '2'.
Thu May 12 13:54:53 2022
 Consumer 2: at iteration 5(consumed). Post-consumption amounts 3 x '1', 0 x '2'
Thu May 12 13:54:53 2022
 Supplier: delivered a '2'. Post delivery amounts 3 x '1', 1 x '2'. 58
Thu May 12 13:54:53 2022
 Consumer 3: at iteration 5(waiting). Current amounts 3 x '1', 1 x '2'
Thu May 12 13:54:53 2022
 Consumer 2 has left.
Thu May 12 13:54:53 2022
 Supplier: read from input a '2'. Current amounts 2 x '1', 0 x '2'.
Thu May 12 13:54:53 2022
 Consumer 3: at iteration 5(consumed). Post-consumption amounts 2 x '1', 0 x '2'
Thu May 12 13:54:53 2022
 Consumer 4: at iteration 5(waiting). Current amounts 2 x '1', 1 x '2'
Thu May 12 13:54:53 2022
 Supplier: delivered a '2'. Post delivery amounts 2 x '1', 1 x '2'. 59
Thu May 12 13:54:53 2022
 Consumer 3 has left.
Thu May 12 13:54:53 2022
 Supplier: read from input a '2'. Current amounts 1 x '1', 0 x '2'.
Thu May 12 13:54:53 2022
 Consumer 4: at iteration 5(consumed). Post-consumption amounts 1 x '1', 0 x '2'
Thu May 12 13:54:53 2022
 Consumer 5: at iteration 5(waiting). Current amounts 1 x '1', 1 x '2'
Thu May 12 13:54:53 2022
 Supplier: delivered a '2'. Post delivery amounts 1 x '1', 1 x '2'. 60
Thu May 12 13:54:53 2022
 Consumer 4 has left.
Thu May 12 13:54:53 2022
 The supplier has left.
Thu May 12 13:54:53 2022
 Consumer 5: at iteration 5(consumed). Post-consumption amounts 0 x '1', 0 x '2'
Thu May 12 13:54:53 2022
 Consumer 5 has left.
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$
```

**Test 3:**

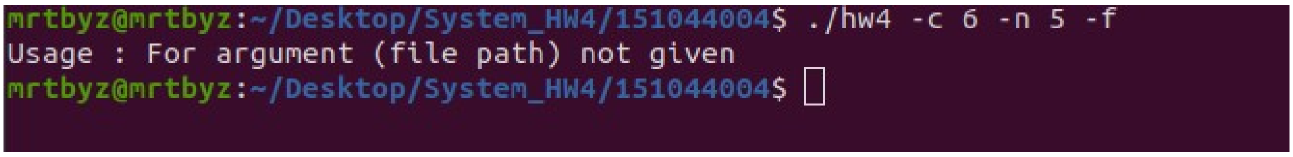
Testing for invalid commands and results shown below:



```
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ ./hw4 -c 3 -n 5 -f test2
Consumer count is invalid
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ 
```



```
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ ./hw4 -c 6 -n 1 -f test2
Iteration count is invalid
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ 
```



```
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ ./hw4 -c 6 -n 5 -f
Usage : For argument (file path) not given
mrtbyz@mrtbyz:~/Desktop/System_HW4/151044004$ 
```