### Write a C program to simulate Earliest deadline first scheduling.

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#define BUFFER\_SIZE 5

#define PRODUCE\_COUNT 10

int buffer[BUFFER\_SIZE];

int in = 0, out = 0;

sem\_t empty, full, mutex;

void\* producer(void\* arg) {

for (int i = 0; i < PRODUCE\_COUNT; i++) {

int item = rand() % 100;

sem\_wait(&empty);

sem\_wait(&mutex);

buffer[in] = item;

printf("Produced: %d at buffer[%d]\n", item, in);

in = (in + 1) % BUFFER\_SIZE;

sem\_post(&mutex);

sem\_post(&full);

sleep(1);

}

pthread\_exit(NULL);

}

void\* consumer(void\* arg) {

for (int i = 0; i < PRODUCE\_COUNT; i++) {

sem\_wait(&full);

sem\_wait(&mutex);

int item = buffer[out];

printf("Consumed: %d from buffer[%d]\n", item, out);

out = (out + 1) % BUFFER\_SIZE;

sem\_post(&mutex);

sem\_post(&empty);

sleep(2);

}

pthread\_exit(NULL);

}

int main() {

pthread\_t prod\_thread, cons\_thread;

sem\_init(&empty, 0, BUFFER\_SIZE);

sem\_init(&full, 0, 0);

sem\_init(&mutex, 0, 1);

pthread\_create(&prod\_thread, NULL, producer, NULL);

pthread\_create(&cons\_thread, NULL, consumer, NULL);

pthread\_join(prod\_thread, NULL);

pthread\_join(cons\_thread, NULL);

sem\_destroy(&empty);

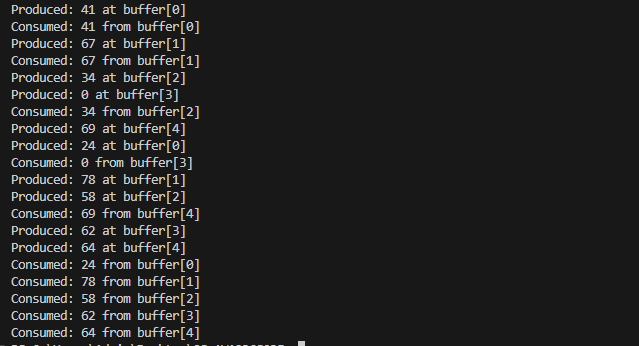
sem\_destroy(&full);

sem\_destroy(&mutex);

return 0;

}

**OUTPUT:**

****