**LAPORAN PRAKTIKUM KOMDATJAR**

**Procedur and Consumer**

Diajukan untuk memenuhi salah satu tugas praktikum Mata Kuliah Komdatjar

Logo, company name

Description automatically generated

**Disusun Oleh:**

**Muhamad Rafli Nur Ikhsan**

**201511048**

**Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika**

**Politeknik Negeri Bandung**

**2022**

\*Mengerjakan Bersama dengan Fitri Fauziyah di satu laptop karena laptop saya tidak bisa menjalankannya

* Script

#include <stdio.h>

#define MaxItems 5 *// Maximum items a producer can produce or a consumer can consume*

#define BufferSize 5 *// Size of the buffer*

sem\_t empty;

sem\_t full;

int in = 0;

int out = 0;

int buffer[BufferSize];

pthread\_mutex\_t mutex;

void \*producer(void \**pno*)

{

int item;

for(int i = 0; i < MaxItems; i++) {

item = rand(); *// Produce an random item*

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

buffer[in] = item;

printf("Producer %d: Insert Item %d at %d\n", \*((int \*)*pno*),buffer[in],in);

in = (in+1)%BufferSize;

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

}

}

void \*consumer(void \**cno*)

{

for(int i = 0; i < MaxItems; i++) {

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

int item = buffer[out];

printf("Consumer %d: Remove Item %d from %d\n",\*((int \*)*cno*),item, out);

out = (out+1)%BufferSize;

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

}

}

int main()

{

pthread\_t pro[5],con[5];

pthread\_mutex\_init(&mutex, NULL);

sem\_init(&empty,0,BufferSize);

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

int a[5] = {1,2,3,4,5}; *//Just used for numbering the producer and consumer*

for(int i = 0; i < 5; i++) {

pthread\_create(&pro[i], NULL, (void \*)producer, (void \*)&a[i]);

}

for(int i = 0; i < 5; i++) {

pthread\_create(&con[i], NULL, (void \*)consumer, (void \*)&a[i]);

}

for(int i = 0; i < 5; i++) {

pthread\_join(pro[i], NULL);

}

for(int i = 0; i < 5; i++) {

pthread\_join(con[i], NULL);

}

pthread\_mutex\_destroy(&mutex);

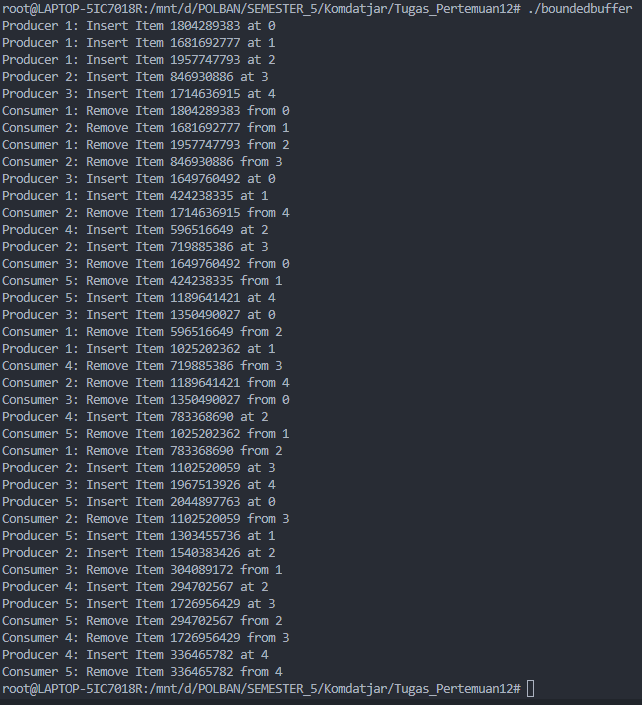
sem\_destroy(&empty);

sem\_destroy(&full);

return 0;

}

* Output



* Lesson Learned

Dalam praktikum kali ini saya belajar procedur-consumer. Saya belajar bahwa solusi tidak memadai akan mengakibatkan deadlock, yaitu kondisi dimana sekumpulan proses diblokir karena setiap proses memegang resource dan menunggu resource lain yang didapatkan oleh beberapa proses lain.