

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ УПРАВЛІННЯ

Звіт з лабораторної роботи № 5
з дисципліни “Основи архітектури ЕОМ та операційні системи”

Виконав: ст. гр. КН-221В

Шулюпов Єгор Русланович

Перевірив: Асистент каф. ПІІТУ

Дмитро Миколайович Ковальчук

Харків

2022

Застосування потоків. Основи роботи з потоками керування

Мета: ознайомитись із засобами синхронізації потоків керування у стандарті POSIX, навчитися компілювати багатопотокові програм серед Linux.

Завдання роботи

Розробити багатопотоковий додаток Linux, який моделює роботу універмагу.

Вимоги до моделі:

1. Універмаг відкрито протягом деякого проміжку часу T . Після цього покупці припиняють з'являтися;
2. В даний час в універмазі можуть знаходитися n людина – решта повинні чекати;
3. Покупці з'являються в універмазі через випадковий проміжок часу від 1 до $r1$;
4. Покупці роблять покупки за випадковий проміжок часу від 1 до $r2$;
5. Перед виходом кожен покупець повинен розрахуватись у касі, яка обслуговує в даний момент часу одну людину - інші повинні чекати;
6. Покупець проводить у касі випадковий проміжок часу від 1 до $r3$. Після розрахунку у касі покупець віддається із системи.

Величини T , n , $r1$, $r2$ та $r3$ повинні задаватися користувачем. Крім цього, має бути можливість задати значення за промовчанням.

Рекомендації:

1. Універмаг і кожен покупець є потоком управління;
2. Вхід універмагу є семафором;
3. Каса представляється мьютексом;
4. Інформація про кожного покупця має супроводжуватися його номером.

```

home > vorpolochek > lab_EOM > #5 > C main2.c > buy()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <pthread.h>
5  #include <semaphore.h>
6  #include <time.h>
7  #include <sys/time.h>
8  pthread_mutex_t mutex;
9  pthread_t thr;
10 sem_t sem;
11 struct timeval tv;
12 int counter = 1, T, n, r1, r2, r3;
13
14 void* buy() {
15     gettimeofday(&tv, NULL);
16     unsigned int seed = (unsigned int)tv.tv_sec;
17     srand(seed);
18     counter++;
19     static int i = 1;
20     int e = i;
21     printf("\nCustomer %d entered the store.\n", e);
22     i++;
23     sem_wait(&sem);
24     srand((unsigned)time(NULL));
25     printf("Customer %d started purchasing.\n", e);
26     int R2 = rand() % (int)r2;
27     sleep(R2);
28     sem_post(&sem);
29     int R3 = rand() % (int)r3;
30     pthread_mutex_lock(&mutex);
31     printf("Customer %d pays for purchases.\n", e);
32     sleep(R3);
33     printf("Customer %d left the store.\n", e);
34
35     printf("Store's open!\n");
36     unsigned int seed = (unsigned int)tv.tv_sec;
37     srand(seed);
38     for (int i = 0; i < T; i++) {
39         int R1 = rand() % (int)r1;
40         sleep(R1);
41         pthread_create(&thr, NULL, buy, NULL);
42     }
43
44     int main() {
45         int value;
46         printf("Choose: \n1 - Default parametrs;\n2 - Your own;\n");
47         scanf("%d", &value);
48         pthread_mutex_init(&mutex, NULL);
49         if (value == 1) {
50             n = T = r1 = r2 = r3 = 3;
51             sem_init(&sem, 0, n);
52             pthread_create(&thr, NULL, store, NULL);
53             if (thr == 0) {
54                 printf("Error\n");
55                 return 0;
56             }
57             pthread_exit(0);
58         } else {
59             printf("Enter amount of time for work: ");
60             scanf("%d", &T);
61             printf("Enter store capacity: ");
62             scanf("%d", &n);
63             printf("Enter r1: ");
64             scanf("%d", &r1);

```

```

74 printf("Enter r2: ");
75 scanf("%d", &r2);
76 printf("Enter r3: ");
77 scanf("%d", &r3);
78 sem_init(&sem, 0, n);
79 pthread_create(&thr, NULL, store, NULL);
80 if (thr == 0) {
81 printf("Error\n");
82 return 0;
83 }
84 pthread_exit(0);
85 }
86 return 0;
87 }

```

```

vorpoloczek@vorpoloczek-VirtualBox: ~/lab_EOM/#6
collect2: error: ld returned 1 exit status
vorpoloczek@vorpoloczek-VirtualBox:~/lab_EOM/#6$ gcc -pthread -o main main5.c
vorpoloczek@vorpoloczek-VirtualBox:~/lab_EOM/#6$ ./main
Choose:
1 - Default parametrs;
2 - Your own;
1
Store's open!
Customer 1 entered the store.
Customer 1 started purchasing.
Customer 1 pays for purchases.

Customer 2 entered the store.
Customer 2 started purchasing.
Customer 1 left the store.
Customer 2 pays for purchases.

Customer 3 entered the store.
Customer 3 started purchasing.
Customer 2 left the store.
Customer 3 pays for purchases.
Customer 3 left the store.

```

Висновок

Під час виконання даної лабораторної роботи ознайомились із засобами синхронізації потоків керування у стандарті POSIX, навчились компілювати багатопотокові програм серед Linux. Також було розроблено багатопотоковий додаток Linux, який моделює роботу універмагу.