#include<stdio.h>

#include<stdlib.h>

#include<Windows.h>

#include<process.h>

#define MAX\_CUSTOMER 100

#define TOTAL\_CLERK 4

volatile LONGLONG Customer\_Number = 0;

volatile LONGLONG Customer\_Serving = 0;

HANDLE Clerk\_Semaphore = NULL;

INT Customer\_Counter[MAX\_CUSTOMER];

UINT \_\_stdcall Clerk(PVOID lp) {

INT Clerk\_Number = \*(INT\*)lp;

while (Customer\_Number < MAX\_CUSTOMER) {

WaitForSingleObject(Clerk\_Semaphore, INFINITE);

INT i = InterlockedIncrement64(&Customer\_Number) - 1;

printf("Serve Customer %2d by Clerk %d (%d)\n", i, Clerk\_Number,

InterlockedIncrement64(&Customer\_Serving));

Sleep(rand()%4);

printf("Serve Customer %2d by Clerk %d done (%d)\n", i, Clerk\_Number,

InterlockedDecrement64(&Customer\_Serving));

ReleaseSemaphore(Clerk\_Semaphore, 1, NULL);

Customer\_Counter[i]++;

}

return 0;

}

int main(int argc, TCHAR\* argv[]) {

HANDLE Clerk\_Handles[TOTAL\_CLERK];

INT Clerk\_Numbers[TOTAL\_CLERK];

srand(GetTickCount());

Clerk\_Semaphore = CreateSemaphore(NULL, TOTAL\_CLERK, TOTAL\_CLERK, NULL);

ZeroMemory(Customer\_Counter, sizeof(Customer\_Counter));

for (int i = 0; i < TOTAL\_CLERK; i++) {

Clerk\_Numbers[i] = i;

Clerk\_Handles[i] = (HANDLE)\_beginthreadex(NULL, 0, Clerk, &Clerk\_Numbers[i], CREATE\_SUSPENDED, NULL);

}

printf("Stuffs are ready\n");

for (int i = 0; i < TOTAL\_CLERK; i++) {

ResumeThread(Clerk\_Handles[i]);

}

WaitForMultipleObjects(TOTAL\_CLERK, Clerk\_Handles, TRUE, INFINITE);

for (int i = 0; i < MAX\_CUSTOMER; i++) {

printf("Customer %2d processed by %d Clerks\n", i, Customer\_Counter[i]);

}

CloseHandle(Clerk\_Semaphore);

system("pause");

return 0;

}