#include<stdio.h>

#include<stdlib.h>

#include<Windows.h>

#include<process.h>

#define MAX\_SUSHI 100

#define BAR\_SIZE 6

#define TOTAL\_CHEF 3

#define TOTAL\_CUSTOMER 5

volatile LONGLONG Sushi\_Number = 0;

volatile LONGLONG Sushi\_on\_Bar = 0;

CRITICAL\_SECTION Sushi\_Bar;

INT head = 0, tail = 0;

INT Sushi\_Counter[MAX\_SUSHI];

UINT \_\_stdcall Customer(PVOID lp) {

INT Customer\_Number = \*(PINT)lp;

while (tail < MAX\_SUSHI) {

EnterCriticalSection(&Sushi\_Bar);

INT i = -1;

if (head > tail) {

i = tail++;

Sushi\_on\_Bar--;

printf("Customer %d Eat Sushi %d (%d)\n", Customer\_Number, i, Sushi\_on\_Bar);

}

LeaveCriticalSection(&Sushi\_Bar);

if (i >= 0) {

Sleep(rand()%4);

Sushi\_Counter[i]++;

}

}

return 0;

}

UINT \_\_stdcall Chef(PVOID lp) {

INT Chef\_Number = lp ? \*((PINT)lp) : 0;

while (head < MAX\_SUSHI) {

EnterCriticalSection(&Sushi\_Bar);

INT i = -1;

if (head - tail < BAR\_SIZE) {

i = head++;

Sushi\_on\_Bar++;

printf("Chef %d makes sushi %d (%d)\n", Chef\_Number, i, Sushi\_on\_Bar);

}

LeaveCriticalSection(&Sushi\_Bar);

if (i >= 0) {

Sleep(rand()%4);

}

}

return 0;

}

INT main(INT argc, PCHAR argv[]) {

HANDLE Chef\_Handles[TOTAL\_CHEF];

INT Chef\_Numbers[TOTAL\_CHEF];

HANDLE Customer\_Handles[TOTAL\_CUSTOMER];

INT Customer\_Numbers[TOTAL\_CUSTOMER];

srand(GetTickCount());

InitializeCriticalSection(&Sushi\_Bar);

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

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

Chef\_Numbers[i] = i;

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

}

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

Customer\_Numbers[i] = i;

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

}

printf("Customers are ready\n");

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

ResumeThread(Customer\_Handles[i]);

}

WaitForMultipleObjects(TOTAL\_CUSTOMER, Customer\_Handles, TRUE, INFINITE);

WaitForMultipleObjects(TOTAL\_CHEF, Chef\_Handles, TRUE, INFINITE);

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

printf("Sushi %2d ate by %d customer\n", i, Sushi\_Counter[i]);

}

DeleteCriticalSection(&Sushi\_Bar);

system("pause");

return 0;

}