Programming 3 (P3): Customer Service Center

Instructor: Dr. Shengquan Wang

ECE 478/CIS 450

Due Time: 10PM, 4/4/2023

Done By: Noureddine Ouelhaci

1-Implementation:

The code implements a simulation of a waiting room with a limited number of chairs and a limited number of service chairs, where customers arrive and wait to be served by one of the assistants. The simulation is implemented using threads and semaphores.

The necessary libraries are included: stdio.h, stdlib.h, pthread.h, semaphore.h, and unistd.h.

Some constants are defined using #define, including NUM_CUSTOMERS, NUM_AWAITING_CHAIRS, NUM_ASSISTANTS, and NUM_SERVICE_CHAIRS.

Three semaphores are declared: awaiting_room_sem, assistants_sem, and service_chairs_sem. awaiting_room_sem is used to limit the number of customers waiting in the room, assistants_sem is used to signal an assistant when a customer is waiting, and service_chairs_sem is used to limit the number of customers being served.

Some variables are declared and initialized, including num_waiting, num_served, next_arrival_time, customer_ids, arrival_times, and service_times. num_waiting keeps track of the number of customers waiting in the room, num_served keeps track of the number of customers who have been served, next_arrival_time keeps track of the next time a customer will arrive, customer_ids is an array of customer IDs, arrival_times is an array of arrival times for each customer, and service_times is an array of service times for each customer.

A customer_thread function is defined. This function takes a pointer to an integer argument, which is the customer ID. The function begins by extracting the arrival

time and service time for the current customer using the arrival_times and service_times arrays.

The function waits for the appropriate amount of time using usleep to simulate the customer's arrival.

The function prints a message to indicate that the customer has arrived.

If there are already NUM_AWAITING_CHAIRS customers waiting in the room, the customer leaves and the function returns.

If there are not already NUM_AWAITING_CHAIRS customers waiting in the room, the function increments num_waiting and signals an assistant by calling sem_post(&assistants_sem).

The function waits for an available service chair by calling sem_wait(&service_chairs_sem).

The function decrements num_waiting and signals that a chair is now available by calling sem_post(&awaiting_room_sem).

The function prints a message to indicate that the customer has started being served.

The function waits for the service to finish by calling usleep for the appropriate amount of time.

The function signals that the service chair is now available by calling sem_post(&service_chairs_sem) and signals an assistant by calling sem_post(&assistants_sem).

The function prints a message to indicate that the customer has finished being served. num_served is incremented to keep track of how many customers have been served.

The customer_thread function returns.

A assistant_thread function is defined. This function takes a pointer to an integer argument, which is the assistant ID. The function runs in a loop until all customers have been served.

The function waits for a customer to arrive or for an assistant to be signaled by calling sem_wait(&assistants_sem).

If there are no customers waiting in the room, the assistant goes to sleep. Otherwise, the assistant begins serving a customer.

Once all customer threads have finished, the main thread signals the assistant threads to exit by calling sem_post(&assistants_sem) NUM_ASSISTANTS times in a loop.

Then, the main thread waits for all assistant threads to finish by calling pthread_join() on each assistant thread in a loop.

Finally, the main thread destroys the semaphores by calling sem_destroy() on each semaphore. The program then returns 0 to indicate successful execution.

2-The algorithm works as follows:

The program initializes three semaphores: awaiting_room_sem, assistants_sem, and service_chairs_sem.

A number of customer threads and assistant threads are created. The number of customer threads is fixed to NUM_CUSTOMERS, while the number of assistant threads is fixed to NUM_ASSISTANTS.

Each customer thread represents a customer. The thread waits for the arrival time, which is specified by the arrival_times array. Once the customer arrives, the thread checks whether there is an available waiting chair in the waiting room. If there is no available waiting chair, the customer leaves. Otherwise, the customer takes a waiting chair, and the number of waiting customers is incremented. The assistant is then signaled that there is a customer waiting. The customer thread waits for an available service chair to become available. Once a service chair is available, the customer thread takes the chair, and the number of waiting customers is decremented. The service time, which is specified by the

service_times array, is waited. Finally, the service chair is released, and the assistant is signaled that a service chair is available.

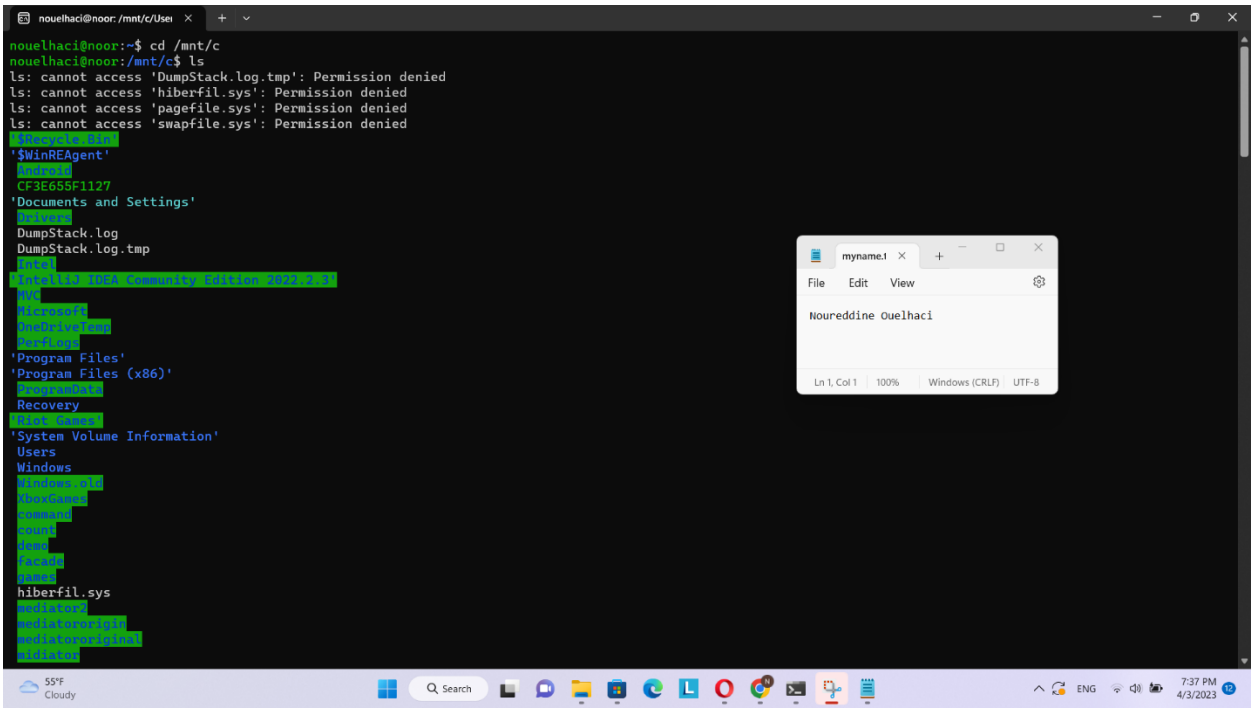
Each assistant thread represents an assistant who serves customers. The thread waits until a customer arrives or until it is signaled that a service chair is available. Once a customer arrives or a service chair is available, the assistant checks whether there is a waiting customer in the waiting room. If there is no waiting customer, the assistant goes to sleep. Otherwise, the assistant serves the waiting customer by taking a service chair, and the customer thread is signaled that the service has started. The assistant thread waits for the service time to finish. Once the service is finished, the assistant releases the service chair, and the customer thread is signaled that the service has finished.

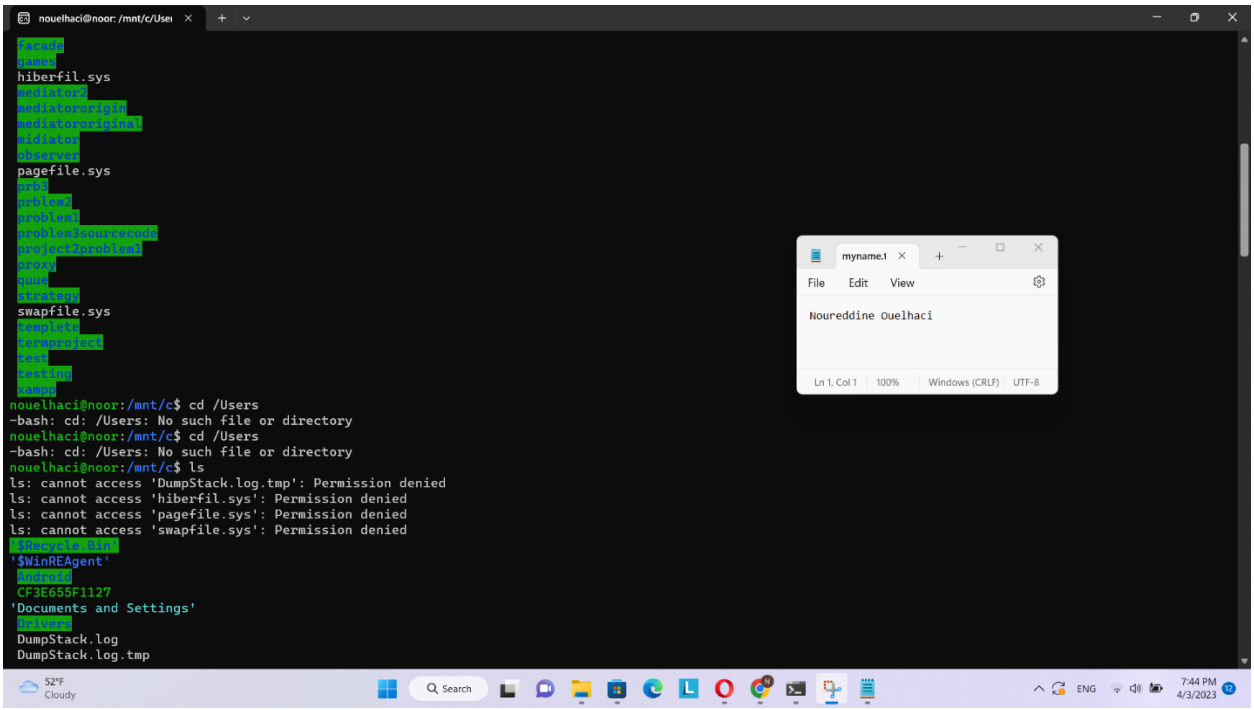
The program waits for all customer threads to finish. Once all customer threads finish, the assistant threads are signaled to exit. The program waits for all assistant threads to finish.

The program destroys the semaphores.

3-Screenshots:

a-file path:


nouelhaci@noor: /mnt/c/Use
nouelhaci@noor:~$ cd /mnt/c
nouelhaci@noor:/mnt/c$ ls
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'pagefile.sys': Permission denied
ls: cannot access 'swapfile.sys': Permission denied
'$Recycle.Bin'
'$WinREAgent'
'$Windows.~BT'
CF3E655F1127
'Documents and Settings'
'Drivers'
DumpStack.log
DumpStack.log.tmp
'Intel'
'IntelliJ IDEA Community Edition 2022.2.3'
'KVS'
'Microsoft'
'OneDriveTemp'
'PerfLogs'
'Program Files'
'Program Files (x86)'
'ProgramData'
Recovery
'Riot Games'
'System Volume Information'
Users
Windows
'Windows.old'
'XboxGames'
'command'
'count'
'demo'
'facade'
'games'
hiberfil.sys
'mediator2'
'mediatororigin'
'mediatororiginal'
'midiator'
myname.txt
File Edit View
Noureddine Ouelhaci
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
55°F
Cloudy
Search
ENG
7:37 PM
4/3/2023


nouelhaci@noor: /mnt/c/Use
'facade'
'games'
hiberfil.sys
'mediator2'
'mediatororigin'
'mediatororiginal'
'midiator'
'showcase'
pagefile.sys
'orb3'
'orblem2'
'orblem1'
'orblem1sourcecode'
'project2problem1'
'proxy'
'quad'
'strategy'
swapfile.sys
'template'
'termproject'
'test'
'testing'
'xampp'
nouelhaci@noor:/mnt/c$ cd /Users
-bash: cd: /Users: No such file or directory
nouelhaci@noor:/mnt/c$ cd /Users
-bash: cd: /Users: No such file or directory
nouelhaci@noor:/mnt/c$ ls
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'pagefile.sys': Permission denied
ls: cannot access 'swapfile.sys': Permission denied
'$Recycle.Bin'
'$WinREAgent'
'$Windows.~BT'
CF3E655F1127
'Documents and Settings'
'Drivers'
DumpStack.log
DumpStack.log.tmp
myname.txt
File Edit View
Noureddine Ouelhaci
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
52°F
Cloudy
Search
ENG
7:44 PM
4/3/2023

```
nouelhaci@noor: /mnt/c/Use
clear
nouelhaci@noor:/mnt/c$ Users
Command 'Users' not found, did you mean:
  command 'users' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>
nouelhaci@noor:/mnt/c$ cd /Users/xv6
-bash: cd: /Users/xv6: No such file or directory
nouelhaci@noor:/mnt/c$ cd \Users\nouel
-bash: cd: Usersnouel: No such file or directory
nouelhaci@noor:/mnt/c$ cd /Users
-bash: cd: /Users: No such file or directory
nouelhaci@noor:/mnt/c$ cd /users/nouel
-bash: cd: /users/nouel: No such file or directory
nouelhaci@noor:/mnt/c$ cd Users/nouel/xv6
nouelhaci@noor:/mnt/c/Users/nouel/xv6$ ls
BUGS            file.c        ls.o          stressfs.d
LICENSE         file.d        ls.sym        stressfs.o
Makefile        file.h        main.c        stressfs.sym
Notes           file.o        main.d        string.c
README          forktest.asm  main.o        string.d
TRICKS          forktest.c    memide.c      string.o
_cat            forktest.d    memlayout.h   swtch.S
_echo           forktest.o    mkdir.asm     swtch.o
_forktest       fs.c          mkdir.c       symlink.patch
_grep           fs.d          mkdir.d       syscall.c
_init           fs.h          mkdir.o       syscall.d
_kill           fs.img        mkdir.sym     syscall.h
_ln             fs.o          mkfs          syscall.o
_ls             gdbutil       mkfs.c        sysfile.c
_mkdir          grep.asm      mmu.h         sysfile.d
_rm             grep.c        mp.c          sysfile.o
_sh             grep.d        mp.d          sysproc.c
_spin           grep.o        mp.h          sysproc.d
_stressfs       grep.sym      mp.o          sysproc.o
_usertests      ide.c         p3.c          timer.c
_wc             ide.d         param.h       timer.d
_zombie         ide.o         picirq.c      timer.o
asm.h           init.asm      picirq.d      toc.ftr
bio.c           init.c        picirq.o      toc.hdr
bio.d           init.d        pipe.c        trap.c
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

52°F Sunset Search ENG 7:45 PM 4/3/2023

```
nouelhaci@noor: /mnt/c/Use
bootmain.c      ioapic.c      proc.d        uart.o
bootmain.d      ioapic.d      proc.h        ulib.c
bootmain.o      ioapic.o      proc.o        ulib.d
buf.h           kalloc.c      rm.asm        ulib.o
cat.asm         kalloc.d      rm.c          umalloc.c
cat.c           kalloc.o      rm.d          umalloc.d
cat.d           kbd.c         rm.o          umalloc.o
cat.o           kbd.d         rm.sym        user.h
cat.sym         kbd.h         runoff        usertests.asm
console.c       kbd.o         runoff.list   usertests.c
console.d       kernel        runoff.spec   usertests.d
console.o       kernel.asm    runoff1       usertests.o
cuth            kernel.ld     sh.asm        usertests.sym
date.h          kernel.sym    sh.c          usys.S
defs.h          kill.asm      sh.d          usys.o
dot-bochsrc     kill.c        sh.o          vectors.S
echo.asm        kill.d        sh.sym        vectors.o
echo.c          kill.o        show1         vectors.pl
echo.d          kill.sym      sign.pl       vm.c
echo.o          lapic.c       sleep1.p      vm.d
echo.sym        lapic.d       spin.asm      vm.o
elf.h           lapic.o       spin.c        wc.asm
entry.S         ln.asm        spin.d        wc.c
entry.o         ln.c          spin.o        wc.d
entryother      ln.d          spin.sym      wc.o
entryother.S    ln.o          spinlock.c    wc.sym
entryother.asm  ln.sym        spinlock.d    x86.h
entryother.d    log.c         spinlock.h    xv6.img
entryother.o    log.d         spinlock.o    zombie.asm
exec.c          log.o         spinp         zombie.c
exec.d          ls.asm        stat.h        zombie.d
exec.o          ls.c          stressfs.asm  zombie.o
fcntl.h         ls.d          stressfs.c    zombie.sym
nouelhaci@noor:/mnt/c/Users/nouel/xv6$ vi p3.c

[3]+  Stopped                 vi p3.c
nouelhaci@noor:/mnt/c/Users/nouel/xv6$
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

52°F Sunset Search ENG 7:45 PM 4/3/2023

b-See the code:

nouelhaci@noor: /mnt/c/user

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define NUM_CUSTOMERS 12
#define NUM_AWAITING_CHAIRS 4
#define NUM_ASSISTANTS 2
#define NUM_SERVICE_CHAIRS 2

sem_t awaiting_room_sem;
sem_t assistants_sem;
sem_t service_chairs_sem;


int num_waiting = 0;
int num_served = 0;
int next_arrival_time = 0;
int customer_ids[NUM_CUSTOMERS] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
int arrival_times[NUM_CUSTOMERS] = {3, 7, 8, 9, 11, 12, 14, 16, 19, 22, 34, 39};
int service_times[NUM_CUSTOMERS] = {15, 10, 8, 5, 12, 4, 8, 14, 7, 2, 9, 3};
int current_start_times[NUM_CUSTOMERS] = {3, 7, 17, 18, 23, 25,0,0, 29, 35, 36, 39};
int current_done_times[NUM_CUSTOMERS] = {18, 17, 25, 23, 35, 29,0,0, 36, 37, 45, 42};

void *customer_thread(void *arg ){
    int cid = *(int *)arg;
    int arrival_time = arrival_times[cid - 1];
    int service_time = service_times[cid - 1];
    int currents_time = current_start_times[cid - 1];
    int currentd_time = current_done_times[cid - 1];

    // Wait until it's time for the customer to arrive
    usleep(arrival_time * 1000000);

    printf("Time %d: Customer %d arrives\n",arrival_time, cid);

    if (num_waiting == NUM_AWAITING_CHAIRS) {
        printf("Time %d: Customer %d                                leaves\n",arrival_time, cid);
```

1,1 Top

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

51°F Rain showers Search ENG 10:28 PM 4/3/2023

nouelhaci@noor: /mnt/c/user

```c
    if (num_waiting == NUM_AWAITING_CHAIRS) {
        printf("Time %d: Customer %d                                leaves\n",arrival_time, cid);
        return NULL;
    }

    // Increment the number of waiting customers and signal the assistants
    sem_wait(&awaiting_room_sem);
    num_waiting++;
    sem_post(&assistants_sem);

    // Wait for an available service chair
    sem_wait(&service_chairs_sem);

    // Decrement the number of waiting customers
    num_waiting--;
    sem_post(&awaiting_room_sem);

    printf("Time %d: Customer %d            starts\n",currents_time, cid);

    // Wait for the service to finish
    usleep(service_time * 1000000);

    // Release the service chair and signal the assistants
    sem_post(&service_chairs_sem);
    sem_post(&assistants_sem);

    printf("Time %d: Customer %d                      done\n",currentd_time, cid);

    num_served++;

    return NULL;
}

void *assistant_thread(void *arg) {
    int assistant_id = *(int *)arg;

    while (num_served < NUM_CUSTOMERS) {
        // Wait for a customer to arrive or for an assistant to be signaled
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

70,0-1 36%

51°F Rain showers Search ENG 10:28 PM 4/3/2023

nouelhaci@noor: /mnt/c/user

```c
void *assistant_thread(void *arg) {
    int assistant_id = *(int *)arg;

    while (num_served < NUM_CUSTOMERS) {
        // Wait for a customer to arrive or for an assistant to be signaled
        sem_wait(&assistants_sem);

        /*if (num_waiting == 0) {
            printf("Assistant %d is going to sleep\n", assistant_id);
        } else {
            printf("Assistant %d is now serving a customer\n", assistant_id);
        }*/

        // Wait for the service to finish or for another assistant to be signaled
        sem_wait(&assistants_sem);
    }

    return NULL;
}

int main() {
    pthread_t customer_threads[NUM_CUSTOMERS];
    pthread_t assistant_threads[NUM_ASSISTANTS];
    int customer_args[NUM_CUSTOMERS];
    int assistant_args[NUM_ASSISTANTS] = {0, 1};

    sem_init(&awaiting_room_sem, 0, NUM_AWAITING_CHAIRS);
    sem_init(&assistants_sem, 0, 0);
    sem_init(&service_chairs_sem, 0, NUM_SERVICE_CHAIRS);

    int i=0;
    int j=0;
    int k=0;
    int l=0;
    int m=0;

        // Create customer threads
    for (i = 0; i < NUM_CUSTOMERS; i++) {
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

103,1 69%

51°F Rain showers Search ENG 10:29 PM 4/3/2023

```
nouelhaci@noor: /mnt/c/user

    int i=0;
    int j=0;
    int k=0;
    int l=0;
    int m=0;

        // Create customer threads
    for (i = 0; i < NUM_CUSTOMERS; i++) {
    customer_args[i] = customer_ids[i];
    pthread_create(&customer_threads[i], NULL, customer_thread, &customer_args[i]);
    }

        // Create assistant threads
    for (j = 0; j < NUM_ASSISTANTS; j++) {
        pthread_create(&assistant_threads[j], NULL, assistant_thread, &assistant_args[j]);
    }

    // Wait for all customer threads to finish
    for (k = 0; k < NUM_CUSTOMERS; k++) {
        pthread_join(customer_threads[k], NULL);
    }

    // Signal the assistants to exit
    for (l = 0; l < NUM_ASSISTANTS; l++) {
        sem_post(&assistants_sem);
    }

    // Wait for all assistant threads to finish
    for (m = 0; m < NUM_ASSISTANTS; m++) {
        pthread_join(assistant_threads[m], NULL);
    }

    // Destroy semaphores
    sem_destroy(&awaiting_room_sem);
    sem_destroy(&assistants_sem);
    sem_destroy(&service_chairs_sem);

    return 0;
}
                                                                                    139,1         Bot
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

c-The steps to find the source code and compile it:

```
root@aa07717e1e0b: /xv6

nouelhaci@noor:~$ docker ps -a
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS                     PORTS                  NAMES
aa07717e1e0b   shqwang/xv6            "bash"                   2 months ago   Exited (255) 7 weeks ago                          xv6cp
4ef317a4a99c   docker/getting-started "/docker-entrypoint.…"   2 months ago   Exited (255) 2 months ago  0.0.0.0:80->80/tcp     admiring_benz
nouelhaci@noor:~$ docker restart xv6cp; docker attach xv6cp
xv6cp
root@aa07717e1e0b:/#
root@aa07717e1e0b:/# cd xv6
root@aa07717e1e0b:/xv6# ls
BUGS            file.c        ls.o          stressfs.d
LICENSE         file.d        ls.sym        stressfs.o
Makefile        file.h        main.c        stressfs.sym
Notes           file.o        main.d        string.c
README          forktest.asm  main.o        string.d
TRICKS          forktest.c    memide.c      string.o
_cat            forktest.d    memlayout.h   swtch.S
_echo           forktest.o    mkdir.asm     swtch.o
_forktest       fs.c          mkdir.c       symlink.patch
_grep           fs.d          mkdir.d       syscall.c
_init           fs.h          mkdir.o       syscall.d
_kill           fs.img        mkdir.sym     syscall.h
_ln             fs.o          mkfs          syscall.o
_ls             gdbutil       mkfs.c        sysfile.c
_mkdir          grep.asm      mmu.h         sysfile.d
_rm             grep.c        mp.c          sysfile.o
_sh             grep.d        mp.d          sysproc.c
_spin           grep.o        mp.h          sysproc.d
_stressfs       grep.sym      mp.o          sysproc.o
_usertests      ide.c         p3.c          timer.c
_wc             ide.d         param.h       timer.d
_zombie         ide.o         picirq.c      timer.o
asm.h           init.asm      picirq.d      toc.ftr
bio.c           init.c        picirq.o      toc.hdr
bio.d           init.d        pipe.c        trap.c
bio.o           init.o        pipe.d        trap.d
bootasm.S       init.sym      pipe.o        trap.o
bootasm.d       initcode      pr.pl         trapasm.S
bootasm.o       initcode.S    printf.c      trapasm.o
bootblock       initcode.asm  printf.d      traps.h
bootblock.asm   initcode.d    printf.o      types.h
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

```
root@aa07717e1e0b: /xv6
_zombie            ide.o          picirq.c       timer.o
asm.h              init.asm       picirq.d       toc.ftr
bio.c              init.c         picirq.o       toc.hdr
bio.d              init.d         pipe.c         trap.c
bio.o              init.o         pipe.d         trap.d
bootasm.S          init.sym       pipe.o         trap.o
bootasm.d          initcode       pr.pl          trapasm.S
bootasm.o          initcode.S     printf.c       trapasm.o
bootblock          initcode.asm   printf.d       traps.h
bootblock.asm      initcode.d     printf.o       types.h
bootblock.o        initcode.o     printpcs       uart.c
bootblockother.o   initcode.out   proc.c         uart.d
bootmain.c         ioapic.c       proc.d         uart.o
bootmain.d         ioapic.d       proc.h         ulib.c
bootmain.o         ioapic.o       proc.o         ulib.d
buf.h              kalloc.c       rm.asm         ulib.o
cat.asm            kalloc.d       rm.c           umalloc.c
cat.c              kalloc.o       rm.d           umalloc.d
cat.d              kbd.c          rm.o           umalloc.o
cat.o              kbd.d          rm.sym         user.h
cat.sym            kbd.h          runoff         usertests.asm
console.c          kbd.o          runoff.list    usertests.c
console.d          kernel         runoff.spec    usertests.d
console.o          kernel.asm     runoff1        usertests.o
cuth               kernel.ld      sh.asm         usertests.sym
date.h             kernel.sym     sh.c           usys.S
defs.h             kill.asm       sh.d           usys.o
dot-bochsrc        kill.c         sh.o           vectors.S
echo.asm           kill.d         sh.sym         vectors.o
echo.c             kill.o         show1          vectors.pl
echo.d             kill.sym       sign.pl        vm.c
echo.o             lapic.c        sleep1.p       vm.d
echo.sym           lapic.d        spin.asm       vm.o
elf.h              lapic.o        spin.c         wc.asm
entry.S            ln.asm         spin.d         wc.c
entry.o            ln.c           spin.o         wc.d
entryother         ln.d           spin.sym       wc.o
entryother.S       ln.o           spinlock.c     wc.sym
entryother.asm     ln.sym         spinlock.d     x86.h
entryother.d       log.c          spinlock.h     xv6.img
```

myname.t

File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

51°F Cloudy Search ENG 8:03 PM 4/3/2023

```
root@aa07717e1e0b: /xv6
bootblock.asm      initcode.d     printf.o       types.h
bootblock.o        initcode.o     printpcs       uart.c
bootblockother.o   initcode.out   proc.c         uart.d
bootmain.c         ioapic.c       proc.d         uart.o
bootmain.d         ioapic.d       proc.h         ulib.c
bootmain.o         ioapic.o       proc.o         ulib.d
buf.h              kalloc.c       rm.asm         ulib.o
cat.asm            kalloc.d       rm.c           umalloc.c
cat.c              kalloc.o       rm.d           umalloc.d
cat.d              kbd.c          rm.o           umalloc.o
cat.o              kbd.d          rm.sym         user.h
cat.sym            kbd.h          runoff         usertests.asm
console.c          kbd.o          runoff.list    usertests.c
console.d          kernel         runoff.spec    usertests.d
console.o          kernel.asm     runoff1        usertests.o
cuth               kernel.ld      sh.asm         usertests.sym
date.h             kernel.sym     sh.c           usys.S
defs.h             kill.asm       sh.d           usys.o
dot-bochsrc        kill.c         sh.o           vectors.S
echo.asm           kill.d         sh.sym         vectors.o
echo.c             kill.o         show1          vectors.pl
echo.d             kill.sym       sign.pl        vm.c
echo.o             lapic.c        sleep1.p       vm.d
echo.sym           lapic.d        spin.asm       vm.o
elf.h              lapic.o        spin.c         wc.asm
entry.S            ln.asm         spin.d         wc.c
entry.o            ln.c           spin.o         wc.d
entryother         ln.d           spin.sym       wc.o
entryother.S       ln.o           spinlock.c     wc.sym
entryother.asm     ln.sym         spinlock.d     x86.h
entryother.d       log.c          spinlock.h     xv6.img
entryother.o       log.d          spinlock.o     zombie.asm
exec.c             log.o          spinp          zombie.c
exec.d             ls.asm         stat.h         zombie.d
exec.o             ls.c           stressfs.asm   zombie.o
fcntl.h            ls.d           stressfs.c     zombie.sym
root@aa07717e1e0b:/xv6# ls p3*
p3.c
root@aa07717e1e0b:/xv6# gcc -o p3 p3.c -lpthread -lrt
root@aa07717e1e0b:/xv6# ./p3
```

myname.t

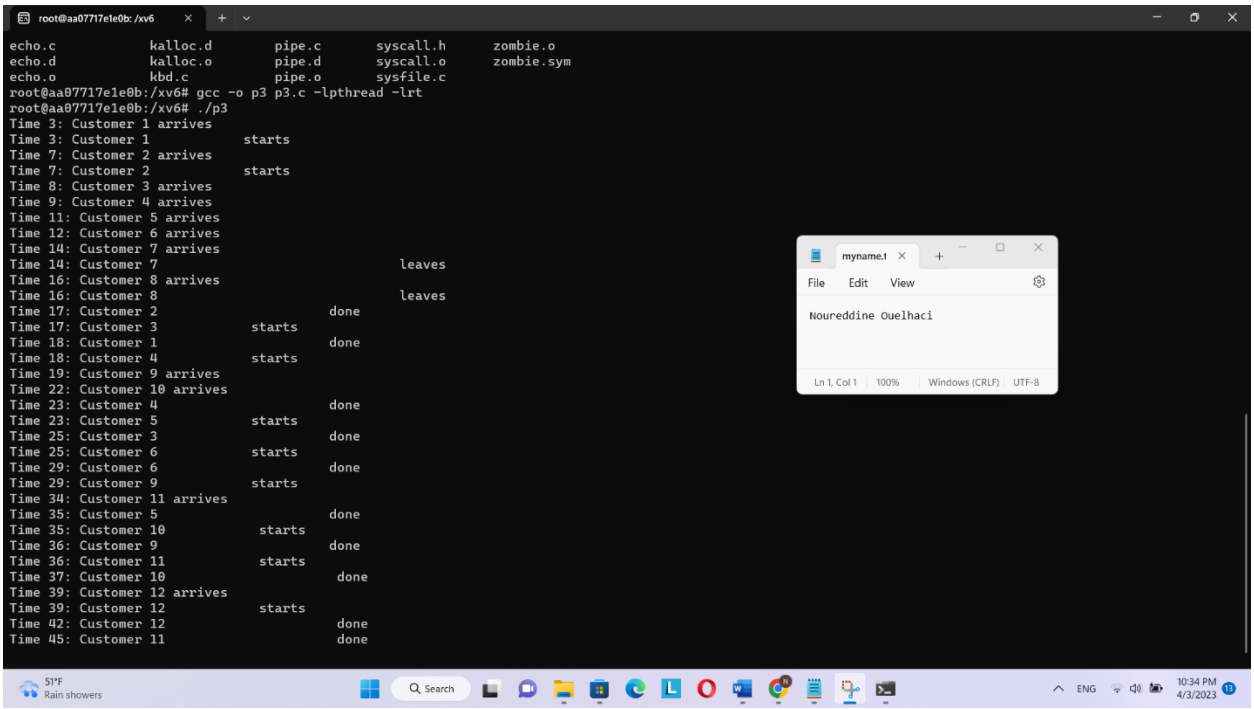
File Edit View

Noureddine Ouelhaci

Ln 1, Col 1 100% Windows (CRLF) UTF-8

51°F Cloudy Search ENG 8:03 PM 4/3/2023

d-The output:



4-the code:

```
#include <stdio.h>

#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define NUM_CUSTOMERS 12
#define NUM_AWAITING_CHAIRS 4
#define NUM_ASSISTANTS 2
#define NUM_SERVICE_CHAIRS 2

sem_t awaiting_room_sem;
sem_t assistants_sem;
sem_t service_chairs_sem;



int num_waiting = 0;
```

```
int num_served = 0;
int next_arrival_time = 0;
int customer_ids[NUM_CUSTOMERS] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
int arrival_times[NUM_CUSTOMERS] = {3, 7, 8, 9, 11, 12, 14, 16, 19, 22, 34, 39};
int service_times[NUM_CUSTOMERS] = {15, 10, 8, 5, 12, 4, 8, 14, 7, 2, 9, 3};
int current_start_times[NUM_CUSTOMERS] = {3, 7, 17, 18, 23, 25,0,0, 29, 35, 36, 39};
int current_done_times[NUM_CUSTOMERS] = {18, 17, 25, 23, 35, 29,0,0, 36, 37, 45, 42};


void *customer_thread(void *arg ){
    int cid = *(int *)arg;
    int arrival_time = arrival_times[cid - 1];
    int service_time = service_times[cid - 1];
    int currents_time = current_start_times[cid - 1];
    int currentd_time = current_done_times[cid - 1];

    // Wait until it's time for the customer to arrive
    usleep(arrival_time * 1000000);

    printf("Time %d: Customer %d arrives\n",arrival_time, cid);

    if (num_waiting == NUM_AWAITING_CHAIRS) {
        printf("Time %d: Customer %d                    leaves\n",arrival_time, cid);
        return NULL;
    }

    // Increment the number of waiting customers and signal the assistants
    sem_wait(&awaiting_room_sem);
    num_waiting++;
    sem_post(&assistants_sem);

    // Wait for an available service chair
    sem_wait(&service_chairs_sem);

    // Decrement the number of waiting customers
```

```
    num_waiting--;
    sem_post(&awaiting_room_sem);

    printf("Time %d: Customer %d           starts\n",currents_time, cid);

    // Wait for the service to finish
    usleep(service_time * 1000000);

    // Release the service chair and signal the assistants
    sem_post(&service_chairs_sem);
    sem_post(&assistants_sem);

    printf("Time %d: Customer %d                    done\n",currentd_time, cid);

    num_served++;

    return NULL;
}

void *assistant_thread(void *arg) {
    int assistant_id = *(int *)arg;

    while (num_served < NUM_CUSTOMERS) {
        // Wait for a customer to arrive or for an assistant to be signaled
        sem_wait(&assistants_sem);

        /*if (num_waiting == 0) {
            printf("Assistant %d is going to sleep\n", assistant_id);
        } else {
            printf("Assistant %d is now serving a customer\n", assistant_id);
        }*/

        // Wait for the service to finish or for another assistant to be signaled
        sem_wait(&assistants_sem);
    }
```

```
    return NULL;
}

int main() {
    pthread_t customer_threads[NUM_CUSTOMERS];
    pthread_t assistant_threads[NUM_ASSISTANTS];
    int customer_args[NUM_CUSTOMERS];
    int assistant_args[NUM_ASSISTANTS] = {0, 1};

    sem_init(&awaiting_room_sem, 0, NUM_AWAITING_CHAIRS);
    sem_init(&assistants_sem, 0, 0);
    sem_init(&service_chairs_sem, 0, NUM_SERVICE_CHAIRS);

    int i=0;
    int j=0;
    int k=0;
    int l=0;
    int m=0;

        // Create customer threads
    for (i = 0; i < NUM_CUSTOMERS; i++) {
    customer_args[i] = customer_ids[i];
    pthread_create(&customer_threads[i], NULL, customer_thread, &customer_args[i]);
    }

        // Create assistant threads
    for (j = 0; j < NUM_ASSISTANTS; j++) {
        pthread_create(&assistant_threads[j], NULL, assistant_thread, &assistant_args[j]);
    }

    // Wait for all customer threads to finish
    for (k = 0; k < NUM_CUSTOMERS; k++) {
        pthread_join(customer_threads[k], NULL);
    }
```

```
    // Signal the assistants to exit
    for (l = 0; l < NUM_ASSISTANTS; l++) {
        sem_post(&assistants_sem);
    }

    // Wait for all assistant threads to finish
    for (m = 0; m < NUM_ASSISTANTS; m++) {
        pthread_join(assistant_threads[m], NULL);
    }

    // Destroy semaphores
    sem_destroy(&awaiting_room_sem);
    sem_destroy(&assistants_sem);
    sem_destroy(&service_chairs_sem);

    return 0;
}
```