Section B

Q1 [15 marks]

Please read the following program and answer the questions.

int x = 0;

void \*function (void \*arg) {

x-=2;

cout << "thread: " << x << endl;

pthread\_exit(NULL);

}

int main () {

// variables are declared here

pid = fork();

x++;

if (pid == 0) {

pthread\_create(&tid, NULL, function, NULL);

pthread\_join(tid, NULL);

x+=1;

cout << "process0: " << x << endl;

}  
else {

wait(NULL);

x+=10;

cout << "process1: " << x << endl;

}

pthread\_exit(NULL);

}

1. List the outputs of the program. Please show the output in order. [7 marks]

Thread : -1

Proess0:-0

Proess1:11

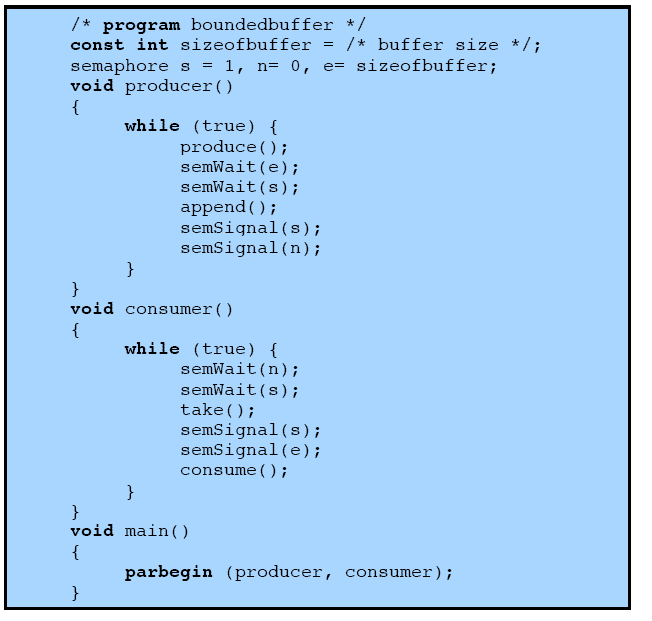
1. Please explain why the program will have the above values. [8 marks]

Threads within the same process have shared access to value. The changed made by one thread is visible to the other thread.

Different processes have separate data and will not affect each other.

Q2 [15 marks]

Refer to the following solution to the bounded-buffer producer/consumer problem using semaphore. The buffer size is 18.



1. What is the value of each of the three semaphores when a consumer wants to take an item from the empty buffer? Assuming there is no producer. [7 marks]

N:-1. S:1. E:18

1. Following (1), a producer is producing data. Show the changes of the three semaphores. [8 marks]

Producing the data: E:17. S:0.

After producing the data: S:1. N:0 unblock the consumer.

Then E 18.

Students might have some details on change of S. This is correct.

Q3 [20 marks]

Consider the following solution using semaphores to solve the one-write many-readers problem.

int readcount;

Semaphore sem\_x, sem\_y;

// Program for the reader

semWait(sem\_y);

readcount++;

if (readcount==1) semWait(sem\_x);

semSignal(sem\_y);

/\*reading\*/

semWait(sem\_y);

readcount--;

if (readcount==0) semSignal(sem\_x);

semSignal(sem\_y);

// Program for the writer

semWait(sem\_x);

/\* writing \*/

semSignal(sem\_x);

A) Suppose only a reader is now reading. No other readers or writer. [14 marks]

(1) [5 marks]

Show the values of the following three variables: readcount, sem\_x, sem\_y

Readcount =1;

Sem\_x=0;

Sem\_y=1;

(2) [5 marks]

What will happen if a writer wants to write, while the first reader in (1) is still reading? Explain with detailed semaphore values.

Sem\_x=-1;

The writer blocked.

(3) [4 marks]

After the writer described in (2), what will happen if two more readers want to read? (The system already has a reader and writer)

They are able to change the readcount and read the data.

B) [6 marks]

(1) [3 marks]

Could we replace a semaphore with a mutex? Explain the possibility.

Sem\_y could be replaced with mutex

(2) [3 marks]

What is the potential issue of this solution?

Readers that join later could read if we have a reader reading

Writers could have potential starvation.