Waled Salem
CSE 460 HW4
SCORE: 65/65

1.Consider the following snapshot of a system:

Process	Allocation	Max	Available
	A B C D	A B C D	A B C D
P0	0 0 1 2	1 0 1 2	2 5 2 0
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm.
a. What is the content of the matrix Need ?
b. Is the system in a safe state? Why?
c. If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted Immediately?

1a)

Need

	A B C D
P0	1 0 0 0
P1	0 7 5 0
P2	1 0 0 2
P3	0 0 2 0
P4	0 6 4 2

1b)

Yes, the system is in a safe state. There exists avaliable resources necessary to run processes P1 and P3. Once these processes are done, the resources will let the remaining processes be completed.

1c)

Yes the state of the system would be:

Process	Allocation	Need	Available
	A B C D	A B C DA B C D	
P0	0 0 1 2	1 0 0 0	2 1 0 0
P1	1 4 2 0	0 3 3 0	
P2	1 3 5 4	1 0 0 2	
P3	0 6 3 2	0 0 2 0	
P4	0 0 1 4	0 6 4 2	

Complete P0:

Process	Allocation	Need	Available
	A B C D	A B C DA B C D	
P1	1 4 2 0	3 3 0	2 1 1 2
P2	1 3 5 4	1 0 0 2	

Waled Salem
CSE 460 HW4
SCORE: 65/65

P3	0 6 3 2	0 0 2 0
P4	0 0 1 4	0 6 4 2

Complete P2:

Process	Allocation	Need	Available
	A B C D	A B C DA B C D	
P1	1 4 2 0	0 3 3 0	3 4 6 6
P3	0 6 3 2	0 0 2 0	
P4	0 0 1 4	0 6 4 2	

Complete P3:

Process	Allocation	Need	Available
	A B C D	A B C DA B C D	
P1	1 4 2 0	0 3 3 0	3 10 9 8
P4	0 0 1 4	0 6 4 2	

Complete P4:

Process	Allocation	Need	Available
	A B C D	A B C DA B C D	
P1	1 4 2 0	0 3 3 0	3 10 10 12

Complete P1

Process	Allocation	Need	Available
	A B C D	A B C DA B C D	
P1			4 14 12 12

2. Consider a swapping system in which memory consists of the following hole sizes in memory order: 16K, 14K, 4K, 20K, 18K, 7K, 9K, 12K, and 15K. Which hole is taken for successive segment requests of
(a) 12K
(b) 10K
(c) 9K
for first fit? Now repeat the question for best fit, worst fit, and next fit.

Waled Salem
CSE 460 HW4
SCORE: 65/65

First Fit:

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	20	18	7	9	12	15

A ⇒H1

H1	H2	H3	H4	H5	H6	H7	H8	H9
4	14	4	20	18	7	9	12	15

B ⇒H2

H1	H2	H3	H4	H5	H6	H7	H8	H9
4	4	4	20	18	7	9	12	15

C ⇒H3

H1	H2	H3	H4	H5	H6	H7	H8	H9
4	4	4	11	18	7	9	12	15

Best Fit:

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	20	18	7	9	12	15

A ⇒H8

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	20	18	7	9	0	15

B ⇒H2

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	4	4	20	18	7	9	0	15

C ⇒H7

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	4	4	20	18	7	0	0	15

Worst Fit:

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	20	18	7	9	12	15

A ⇒H4

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	8	18	7	9	12	15

B ⇒H5

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	8	8	7	9	12	15

Waled Salem
CSE 460 HW4
SCORE: 65/65

C ⇒H1

H1	H2	H3	H4	H5	H6	H7	H8	H9
5	14	4	8	18	7	9	12	15

Next Fit:

H1	H2	H3	H4	H5	H6	H7	H8	H9
16	14	4	20	18	7	9	12	15

A ⇒H1

H1	H2	H3	H4	H5	H6	H7	H8	H9
4	14	4	20	18	7	9	12	15

B ⇒H2

H1	H2	H3	H4	H5	H6	H7	H8	H9
4	4	4	20	18	7	9	12	15

C ⇒H4

H1	H2	H3	H4	H5	H6	H7	H8	H9
4	4	4	11	18	7	9	12	15

3.

A.The Virtual Address is 20. The slot is 0k-4k of the address table, which can be translated to 4k-8k on the physical space. Result:

8k+20
= 8 *1024 + 20
= 8212

B. The address is 4100, which is translated to the 0k-4k on the physical address table
Result

4k + 4100 - 4k
= 4100

C. the address is 8300. Which is translated to 8k on the physical space. Result:

24k+(8300-8k)
= 24684

4.

Code:

Waled Salem
CSE 460 HW4
SCORE: 65/65

```cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <SDL/SDL.h>
#include <SDL/SDL_thread.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

using namespace std;

SDL_mutex *mutex;
SDL_cond *reader_queue;
SDL_cond *writer_queue;
int readers = 0;
int writers = 0;
int reader(void* data) {
    while(true) {
        SDL_Delay(rand() % 1000);
        SDL_LockMutex(mutex);

        while(writers != 0)
            SDL_CondWait(reader_queue, mutex);

        ++readers;
        SDL_UnlockMutex(mutex);
        ifstream file("counter.txt");

        if(file.good()) {
            int counter;
            file >> counter;
            cout << *((string*) data) << " with counter value " << counter << endl;
            file.close();
        } else
            cout << *((string*)data) << " unable to read counter" << endl;

        SDL_LockMutex(mutex);

        if(--readers == 0)
            SDL_CondSignal(writer_queue);

        SDL_UnlockMutex(mutex);
    }
}

int writer(void* data) {

    while(true) {
        SDL_Delay(rand() % 1000);
        SDL_LockMutex(mutex);

        while(writers != 0 and readers != 0)
            SDL_CondWait(writer_queue, mutex);

```

Waled Salem
CSE 460 HW4
SCORE: 65/65

```
52          while(writers != 0 and readers != 0)
53              SDL_CondWait(writer_queue, mutex);
54
55          ++writers;
56          SDL_UnlockMutex(mutex);
57          int counter = -1;
58          ifstream file("counter.txt");
59
60          if(file.good()) {
61              file >> counter;
62              file.close();
63          }
64
65          ++counter;
66          ofstream file_write("counter.txt", std::ofstream::trunc);
67          file_write << counter;
68          file_write.close();
69
70          cout << *((string*)data) << "with counter value " << counter << endl;
71          SDL_LockMutex(mutex);
72          --writers;
73          SDL_CondSignal(writer_queue);
74          SDL_CondSignal(reader_queue);
75          SDL_UnlockMutex(mutex);
76          }
77  }
78
79  int main() {
80      SDL_Thread* reader_thread[20];
81      SDL_Thread* writer_thread[3];
82
83      mutex = SDL_CreateMutex();
84      reader_queue = SDL_CreateCond();
85      writer_queue = SDL_CreateCond();
86
87      for(int i = 0; i < 3; i++) {
88          stringstream ss;
89          ss << "writer " << i;
90          string* name = new string(ss.str());
91          writer_thread[i] = SDL_CreateThread(writer, (void*) name);
92      }
93
94      for(int i = 0; i < 20; i++) {
95          stringstream ss;
96          ss << "reader " << i;
97          string* name = new string(ss.str());
98          reader_thread[i] = SDL_CreateThread(reader, (void*)name);
99      }
100
101     SDL_DestroyCond(reader_queue);
102     SDL_DestroyCond(writer_queue);
103     SDL_DestroyMutex(mutex);
104     return 0;
105 }
```

C++ ▾ Tab Width: 4 ▾ Ln 41, Col 1

Output:

Waled Salem
CSE 460 HW4
SCORE: 65/65

```
reader 15 with counter value 7
reader 7 with counter value 7
reader 18 with counter value 7
reader 6 with counter value 7
reader 14 with counter value 7
reader 4 with counter value 7
reader 13 with counter value 7
reader 4 with counter value 7
reader 19 with counter value 7
reader 15 with counter value 7
reader 16 with counter value 7
writer 2 with counter value 8
writer 1 with counter value 9
reader 18 with counter value 9
reader 8 with counter value 9
reader 7 with counter value 9
reader 17 with counter value 9
reader 4 with counter value 9
reader 14 with counter value 9
reader 13 with counter value 9
reader 11 with counter value 9
reader 1 with counter value 9
reader 0 with counter value 9
writer 1 with counter value 10
reader 10 with counter value 10
```

I believe I have earned a perfect score of **65/65** since I have finished all four of the required tasks**.**