CSC 8980 Distributed Systems Name: Ishu Goyal

Assignment – 1

Question 1:

a) There are currently 31 Global Positioning System (GPS) satellites developed and operated by the United States. These satellites are present in six different orbital planes which allows for six to nine satellites to be visible from a given point at any time. Out of these 31 satellites 24 to 25 satellites are active satellites and rest are present as backup satellites in each orbit.

b) Four satellites are needed to determine location on the Earth using the GPS:
three to determine a position on the Earth, and fourth satellite is used to solve an equation that lets it determine the exact time, without needing an atomic clock.

c) Fourth satellite measurements are used to solve equation that lets it determine the exact time, thereby eliminating the need of atomic clock. Therefore, the receiver uses four satellites to compute latitude, longitude, altitude, and time.

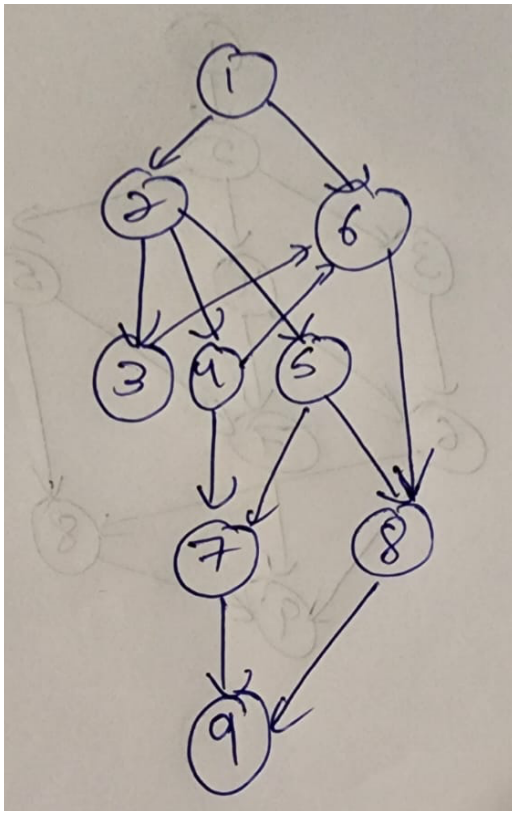
d) **Satellite clock synchronisation:**
The satellite clocks are synchronized regularly by an associated ground station of the provider.

Receiver clock synchronisation:
GPS applications are dependent on the travel time of the signal. The travel time of signal is need for calculation of position using four satellites Three satellites are used for determination of the position in 3D and fourth satellite is for the receiver clock error which is the synchronization between the receiver and the satellite.

e) The Wide Area Augmentation System (WAAS) is an air navigation aid developed by the Federal Aviation Administration to augment the Global Positioning System (GPS), with the goal of improving its accuracy, integrity, and availability. Essentially, WAAS is intended to enable aircraft to rely on GPS for all phases of flight, including precision approaches to any airport within its coverage area. It may be further enhanced with the Local Area Augmentation System (LAAS) also known by the preferred ICAO term Ground-Based Augmentation System (GBAS) in critical areas.

f) Military grade receivers use two GPS frequencies whereas general purpose receivers use one frequency. Using two GPS frequencies improves accuracy by correcting signal distortions caused by Earth's atmosphere. Dual-frequency GPS equipment is commercially available for civilian use, but its cost and size has limited it to professional application

Question 2:

(a) 

(b) Following Processes can run concurrently:
   a. 5,6
   b. 7,8
   c. 4,5
   d. 4,1,3

(c) To make sure above precedence relationship is deterministic, we can first find the transitive closure of precedence relation and then remove the process pairs from the transitive closure for which the following condition fails :

$(D(p_i) \cap R(p_j)) \cup (D(p_j) \cap R(p_i)) \cup (R(p_i) \cap R(p_j)) = \emptyset$

At the end MPS would be obtained which is Deterministic system of processes.

(d) To find the MPS:

first find the transitive closure of the given precedence relation:

Transitive closure =
{(1,2),(1,3),(1,6),(2,3),(2,4),(2,5),(2,6)(2,7),(2,8),(2,9),(3,6),(3,8),(3,9),(4,6),(4,7) ,(4,8),(4,9),
(5,7),(5,8),(6,8),(6,9),(7,9),(8,9)}

Second check for processes which can be executed concurrently from the transitive closure :

Find pairs from transitive closure that satifies this condition:
$(D(p_i) \cap R(p_j)) \cup (D(p_j) \cap R(p_i)) \cup (R(p_i) \cap R(p_j)) = \emptyset$

Based on this condition the following is MPS:

➔ ' ={(1,2),(1,6),(2,5),(2,6),(2,8), (3,8),(3,9),(4,7) ,(4,8), (5,7), (6,9), (8,9)}

Question 3:

Theorem: a mutually non-interfering system of processes is determinate.

A system is called determinate if the sequence of values written in each memory cell is the same for any order of execution allowed by P.

To decide if a system of processes is determinate or note

Read Set R(Pi) is the set of variables referenced during the execution of process (statement) Pi that were not written by Pi.

Write Set W(Pi) is the set of variables referenced during the execution of process Pi that were written by Pi

For the program at the start of the section, there is:

R(read (a))

R(c = a +b)

W(read (a))

W(c = a +b)

Let Pi and Pj be two processes.

Pi and Pj can be executed concurrently if they satisfy following Bernstein conditions

$R(Pi) \cap W(Pj) = \emptyset$

$W(Pi) \cap R(Pj) = \emptyset$

$W(Pi) \cap W(Pj) = \emptyset$

- A determinate system of processes is a system of processes P for which each element of domain(P) produces the same set range(P)
- A mutually noninterfering system of processes is a system of processes P in which all pairs of processes meet the Bernstein conditions.
- Processes p and q are noninterfering if either process is a predecessor of the other, or the processes satisfy the Bernstein conditions.
- The initiation of a process p is written p, and the termination of p is written p.

So, if a system is mutually non-interfering, it is determinate.

Hence Proved, a mutually non-interfering system of processes is determinate.

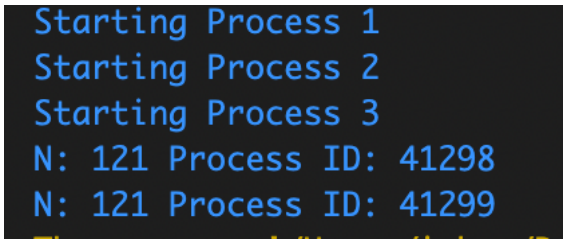
Question 4:

Sourcecode:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

const char * file_name = "num.txt";

void process() {
   pid_t pid; //Process ID
   FILE * file = fopen(file_name, "r"); //Open file
   int N = 0;
   fscanf(file, "%d", & N); //Read Number
   fclose(file); //Close file
   if ((pid = getpid()) < 0) { //Get Process ID
      printf("unable to get pid");
   } else {
      printf("N: %d Process ID: %d", N, pid);
      printf("\n");

   }
   file = fopen(file_name, "w");
   N++;
   putw(N, file);
   fflush(file);
   fclose(file);
}
int main() {
   if (fork() == 0) {
      printf("Starting Process A\n");
      process();
   }
   if (fork() == 0) {
      printf("Starting Process B\n");
      process();
   }
   if (fork() == 0) {
      printf("Starting Process C\n");
      process();
   }
   return 0;
};
```

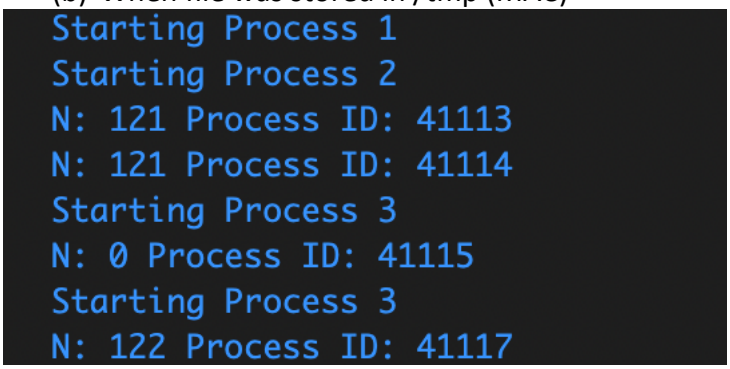
Output:

(a) When file was stored on local disk



Based on the output above I feel that the three process are getting triggered but due to some kind of race condition two of the processes gets access to file and one of the processes was able to complete the set of instructions and by the time others to reach the file is empty as it gets flushed.

(b) When file was stored in /tmp (MAC)



All the three processes were able to access the file and complete its set of instructions.

The difference I felt was on local disk, it asked for permissions to apply changes to file which might have caused delay for process 3 and in tmp directory it didn't asked for permissions and all three processes were able to execute their set of instructions