Exercise -5 :-

Q-1 :-

The top of the list for the worst software and hardware computer engineering design flaw and/or software defect of all time would be the Challenger and Columbia Shuttle Loss

The first disaster happened was the challenger Space Shuttle on January 28th , 1986. Only after 73 seconds of the lift off it fell apart and took the life of 7 people which were inside the space shuttle . Out of 7 , 5 were NASA astronauts. While the lift off took place there was failure in the Solid rocket boosters (SRBs). There was a low temperature in the impairment of an O-ring.

O-ring is basically a critical seal used in between the segments of SRB casings. As a result of this hot air gases went through the booster sections and burn through external tank. Thus it exploded. This problem was not faced foe the first time while the mission was running, it actually happened at the ground also when the engineers warned the managers of NASA about the O-rings safety when it went below 12 C.

Thus we can say that it was the fault of the organisation mainly and not the fault of the engineers on the first place which resulted the life of 7 people.

Another incident happened on Feb 1 2003, when Columbia Space Shuttle disintegrated. This also resulted in the loss of 7 crew members. This incident happened during the landing of the space shuttle. Initially there were loss of temperature sensors on the left wing which resulted in tire pressure indications on the left main gear. The thermal protection layer unit requested that the astronauts should leave Columbia so that they can look for the damage possible but the DOD refused their request and did not allow the astronauts for a spacewalk thus we can say again that it was not the fault of engineers for the failure but the management.

2) NORAD :-

This incident of false alarms is 2nd in my list because it did not take the lives of any people. It happened 3 times when the NORAD system failed. NORAD is North American Aero Space Defense Command. It is joint organisation of America and Canada to protect the borders from any aero space threat. On 9th November 1979, the first false alarm was generated when the technician loaded the test tape but failed to put it as a "test" and as a result the warnings were spread across the globe for a nuclear attack from Russia. Everyone was alerted the fighter jets were ready with the nuclear bombs. Though almost every post knew it was a false warning since all of the posts did not generate the warning.

On 3rd June and 6th June 1980 the same problem was generated but this time it was the coputer problem which failed and generated the warnings.

After the 9/11 attacks the has been expanded a lot and they have been tracking even a small aircraft entering the USA or Canada borders now , which is even helpful to them for catching drug trafficking.

Though these mistakes were made but still USA did not launch any nuclear attack but it can result in the attack of nuclear bombs on different countries and a war could be started if the situation is not evaluated very carefully.

3) Blackberry :-

This is 3rd in my list because it did not involve any problem to people's lives. It was mainly a technicality problem.

Blackberry launched 9500 series of smartphone. They launched the first mobile without a physical keyboard. They used a button via surepress which they used to navigate through the screen for the available keyboard letters.

This created a storm and sold 500,000 units in just 1 month in January 2009. In the initial phase they did not receive any complaints but after sometime there were some mixed reviews regarding this smartphone. Some said that the button did not work perfectly while some said it was working fine.

It got very popular in different countries and inspite of the issues and complaints by the customer the company were selling the products without testing it more and haulting the production. The managers of the company neglected the complaints and eventually the problem increases worldwide.

As a result of which Verizon had to replace almost all the smartphones and they had to bear a loss of 500 Million US Dollars. If the company had verified this issue before putting it in the market and spend some more time in testing of the smartphone then they could have saved a lot of money on the first place itself.

Q-2

1) **Therac-25:-**

   It was a radiation therapy machine produced by Atomic Energy of Canada Limited in 1982. It had involved 6 accidents after that from the period of 1985 and 1987. These accidents involved sheer negligence of the engineers in the software of the program which resulted in massive radioactive doses given to the patients which was 100 times stronger than the normal one, which was given. This resulted in the lives of 3 people in the end due to radiation overdose.
   The first accident caused because the engineer accidently gave the X-rays direclt to the patients before converting it into the electron mode.
   Second accident took place when the electron beam to activate during field-light mode, during which no beam scanner was active or target was in place.This incident would not have taken place if they were not so dependent on the software for the safety check and had kept the inter locks at the hardware level for the safety.
   After these incidents patients were seen with radiation effects on them because the beam was delievered to them 100 times and that too to a narrower surface area. Since the surface Area was smaller they were affected more.

   At the time of checking what went wrong in these accidents it was clearly mentioned that the fault was not of some random coding mistakes but just a poor design and implementation practices.

Some of the root causes were :-

1) AECL did not review the software code independently.
2) AECL did not consider the failure modes the machine would act to
3) The system noticed that there was some problem and halted the procedure but since no error codes were written in the manual the operator just continued by pressing “p”
4) No one believed that there were complaints because of sheer over confidence.
5) Until and unless it was assembled at the hospital, AECL never tested the hardware and the software together.

The software which they wrote for the system was basically written in assembly language. This needed more attention for testing and good design.

2)**Three Mile Island Unit:-**

It is a nuclear power plant which has 2 units , TM1 and TM2. The TMI-2 suffered a massive meltdown problem on 28th March 1979. Though this incident did not kill any people but it could have thus It is 2nd in my list. After this incident it was closed and removed from the site

Another incident happened on March 28th 1979, due to malfunction of cooling system. It resulted in partial meltdown of reactor core. This occurred due to two systems malfunctioning and not meeting the deadline of their respective tasks. It resulted in exposure to 2 million people. The pennysylvania department kept a record of 30,000 people that lived within 5 miles radius to check if they are getting affected from the radiation or not. They kept the record for 20 years and once they did not find any results of radiation in them they started recording them.

This could have been stopped by proper software analysis of the product and taking care of the deadlines when there is a parall processing of the system going on.

3)Toyota Vehicle Recalls:-

This incident is 3rd in my list. The problem with the vehicles of Toyota was the ABS (Anti Braking System). It resulted in total death of 37 people. When the problem came with prius brakes, Toyota said that it was because of the software glitch in the system. But the story is different when NHTSA came into action. They said that the issue was the "short delay" in regenerative braking when hitting a bump, resulting in increased stopping distance. This might be happening because of a failure in the real time system failure which they might have ignored in the beginning stage of the launch, which resulted in changing the software of ABS of 133,000 Prius vehicles in the U.S. and 52,000 in Europe This was solved by Toyota on Feb 6 2010 .

Q-3 :-

Key findings :-

According to the paper provided, we know that the NAVY began running shipboard applications under Windows NT ,this step was taken as a cost reduction step. They wanted to function most of the things with the help of the software and very less with the help of man and this software helped them with that. But as we know all automations face some or the other problem sometime in their lifespan.
This system also showed some bad results due to bad data fed into the computers. Due to such an event the Yorktown was dead in the water for about 2 hours and 45 minutes on May 2 1997 when the software was frozen.
The Yorktown lost control of its propulsion system because its computers were unable to divide by the number zero, this was said by the memo. It was not able to divide by 0, because of the fault of system administrator because he entered 0 into the data field for remote data base manager program. Thus it crashed all the LAN consoles and miniature remote terminal units.
After the result of these mishap, the officals suggested that Unix is much better operating system than NT. Because the WINDOWS NT was not able to solve a simple problem which was even solved by calculator and thus if they preferred a Unix system than the chances of solving the errors were high and thus we could have saved a lot of Money which was spent after Windows NT and the probability of error could have been decreased.
These results were a narration from Gregory but according to Redman , Gregory's findings were not related. According to him, the systems cannot be blamed and Gregory just wanted a long time employment that's why he made such accusations on the system.

Alternate Key Findings :-

In the paper found on references on D2L, about the LAN Crashes & Smart chip system, A system administrator fed the wrong data into the system causing the system to overflow because it was not able to divide by 0 and all the LANs got crashed.
The normal project time which was supposed to be 3 years for such a huge project was completed in just 8 month and according to redman it was just a political pressure for completing the project. It was the first ship in the navy that was commissioned without any prior testing of the systems and when engineers also reported that they also reported that some wiring was left over.
According to Gregory NT system was responsible for the damage that was made but according to Rushton NT was not at all responsible for the damage caused that day.
Rushton believes that NT system is very essential to future ship system designs such as Smart ship program. He also believed that it was the problem of the codes and bad software structure.
Navy officials a said that Windows NT is not useful for emergency systems and Unix will be better in that case.

According to **John Kirch**, https://www.wired.com/1998/07/sunk-by-windows-nt/

Windows NT is the best and there is no match for an Unix system to that. Mocrosoft's Bill gates also supported for the smart ship program

Moreover, http://www.linuxfocus.org/English/May1998/article41.html ,in the link provided if we take views of MIS Professionals, they also support the use of Windows NT over Unix.

C)

Based on my understanding , I conclude that it was operator's mistake but not completely. Operator did enter 0 at the place where he should not have entered and as a result the system crashed. But it is not his fault completely since in the time of hustle, it is the human tendency to make mistakes. I put some blame in the coding of the Windows NT system which was used in this machine because this type pf precaution should have been taken while designing the software for the machine in which the government has spent billions of dollars. Such type of mistake is not at all allowed when so much things are at stake. This problem of dividing by 0 is even solved in a small calculator then how come it was not over looked into such a huge navy ship. Another blame I would put is on the company who developed the system In just 8 months when such a system would have taken nearly 3 years to built. If the ship was tested properly and some time was spent on looking at the wiring of the ship and coding of the software then this problem could have been solved.

D) file:///C:/Users/MGWork/Downloads/Redhawk-Linux-brochure.pdf

Point 1 :- According to the pdf shown above, Redhawk system is more helpful in the critical missions like this . It is well tested as compared to Windows NT. It provides single kernel environment that only controls all the functions. It is not dependent on any other environment. For such Critical missions we need high speed file I/O, networking and graphics with real time scheduling. Only RedHawk system can do this type of functions on a very high speed.

Point 2 :- Multithreading & Pre emption :-

RedHawk Linux allows many threads to work at the same time and parallel processing can be done in the kernel simultaneously. To protect it from any possible security fraud, kernel uses the semaphores and spinlocks. In this it also has the benefit that the kernel can transfer the control from the lower priority process to a higher priority process. But if the lower priority process is in the kernel critical section. Due to this facility, it can respond immediately to an external process very rapidly.

Due to these features of RedHawk Linux, I would conclude that we should use this system as compared to any Windows or traditional RTOS or Cyclic Executive.

Q-4

Project Proposal :-

Time Lapse Image Acquisition :-

Objective of the project Time Lapse :- It is a technique of photography, when the frequency at which film frames are captured is much lower than what we are seeing in the sequence (Wikipedia). The aim of doing time lapse is to use the scheduling algorithms which are real time. For example, I capture the frames at a slower rate and then those captured frames I run at the faster rate. The video formed with faster rate is called Time lapse video.

Lets say I want to see the sun rise and sunset of a particular day in just a minute and know the shades of the sky at that time and I dont have the whole day to waste after it, then I will start recording the camera from sunrise till sunset and then will watch the time lapse of that ,with a constant frame rate , thus I will be able to see the sunrise and sunset in a particular time frame. So we can say that time lapse is basically used to detect the slow changes in a fast manner.

Here it is very important that only after the completion of one frame 2nd frame should be started or else I will miss the change that happened in that particular frame.

Components Used in this Project :-
1) Raspberry – Pi -with Keyboard and Mouse
2) Logitech C200 camera

Minimum Requirements :-
1)My resolution will be of 640*480 for this project
2)My every frame will be captured at 1Hz rate that means at every 1 sec 1 frame will be captured by the camera.
3)If a capture 2000 frames means it will be captured in 2000 sec (33 min ,20 sec) , which I need to show in around 1-2 minutes. The frames captured in 2000 sec I will be able to see in nearly 1-2 minutes.

Real time scheduling methods can be used for this requirements.

For Target Goals :-
1) I will first of all take the 2000 frames then I will compress the frames. By compressing the frames I will be able to reduce the space which is used. So that I can send it over the ethernet.
2)I am planning on using the ffmpeg compression technique for doing time frame of many images. It a lossy method of **compression.**
3) Once the compression is done I will **store** it and transfer through ethernet. Once transfered through ethernet I will again enhance it and play it back at a faster speed on the other end.
4) I will try to reduce the jitter in these frames captured so that my final video is clear and continuous.

For Stretch Goals :-

1)I will first of all try to get a higher frame rate at 10 Hz continuous download upto 6000 frames and check jitter in the system.

If time permits , I am planning on doing one of the extra elements which requires background elimination, sharpening of the iamge, detection of a target and computation of its centroid, segmentation of the image, statistics such as a spectral histogram computed for the image.
- For background reduction I am planning on using a gaussian mixture based Background/Foreground Segmentation algorithm.
Mainly I will be using opencv functions and algorithms for this purposes.

For calculating the centroid , I will first get all the black pixels, After getting that I will take the average of all coordinates of black pixel.

Software :-
1) I will use a linux platform R-pi 3 B+ for this and use opencv 3 on that.
2) i will use p threads for different services
3) Will use semaphores
4)Time stamp will be used
Flow chart Steps will be as follows :-
1) Get the raw data of specific amount of frames
2) Save those images
3) Compress the images
4)Timestamp
5)transfer through ethernet
6) Play Video

Goal with timetable:-
1) My first Goal will be to complete the requirements by 4th august
2)By 10th august i will finish target and stretch Goals
3)By 13th August I will try to complete extra element of reducing background etc.
4) If my first 2 steps are not completed by 10th august I will spend my time in completing them first by 13th august and will not do the extra elements.

References :-
1)https://en.wikipedia.org/wiki/North_American_Aerospace_Defense_Command#False_alarms

2)https://en.wikipedia.org/wiki/Space_Shuttle#Shuttle_disasters

3)https://en.wikipedia.org/wiki/BlackBerry_Storm#Sales_and_replacement

4)https://en.wikipedia.org/wiki/Therac-25

5)https://en.wikipedia.org/wiki/Three_Mile_Island_Nuclear_Generating_Station

6)https://en.wikipedia.org/wiki/2009%E2%80%9311_Toyota_vehicle_recalls#Anti-lock_brake_software_recall

7)http://ecee.colorado.edu/%7Eecen5623/ecen/labs/Linux/extended_lab_requirements.html