BVRIT HYDERABAD College of Engineering for Women
Department of Electronics and Communication Engineering

Hand Out

Subject Name: Embedded System Design

Prepared by (Faculty(s) Name):
1. Dr. M. Parvathi, Professor, ECE

Year and Sem, Department: IV Year-I Sem, ECE

Unit – I: Introduction to Embedded Systems

Important Points / Definitions: (Minimum 15 to 20 Points covering complete topics in that unit)

- An embedded system is a combination of 3 things:
  a. Hardware
  b. Software
  c. Mechanical Components
  And it is supposed to do one specific task only.

- An embedded system is designed to do a specific job only. Example: a washing machine can only wash clothes, an air conditioner can control the temperature in the room in which it is placed.

- An embedded system will always have a chip (either microprocessor or microcontroller) that has the code or software which drives the system.

- The embedded system is designed to perform a specific task whereas as per definition the general-purpose computer is meant for general use. It can be used for playing games, watching movies, creating software, work on documents or spreadsheets etc.
- The classification of embedded system is based on following criteria's such as based on generation, complexity & performance, deterministic behaviour, triggering etc.
- The application areas and the products in the embedded domain are countless and are consumer electronics, household appliances, automotive industry, home automation & security systems, telecom, computer peripherals, computer networking systems, healthcare, and banking & retail, etc.

Questions (Minimum 6 to 8)

1. Define Embedded System with the help of Microwave Owen as an example ?
2. Differentiate between general purpose computers & embedded systems
3. Give a classification of embedded systems
4. List some applications of embedded systems
5. Explain the various possible purposes of using and embedded system.
6. Give examples for latest embedded devices.

Fill in the blanks / choose the Best: (Minimum 10 to 15 with Answers)

1. An embedded system is a combination of ---------- [b]
[a] hardware, software, memory [b] hardware, firmware, mechanical components
[c] hardware, system bus, memory [d], cpu, i/o and memory

2. An embedded system can do ----------task [c]
[a] multi task at a time [b] single task at a time
[c] specific task [d] all the above.

3. The first recognized embedded system is the ----------developed by MIT lab. [a]
[a] Apollo Guidance Computer(AGC) [b] Apollo Satellite Computer(ASC)
[c] Apollo Automatic Glider (AAG) [d] Apollo Guidance Satellite (AGS)

4. Embedded systems are ---------- [b]
[a] general purpose [b] special purpose
[c] domain purposes [d] all the above

5. An embedded system must have [b]
(a) hard disk (b) processor and memory
(c) operating system (d) processor and input-output unit(s).

6. An embedded system hardware can [a]
(a) have microprocessor or microcontroller or single purpose processor (b) have digital signal processor (c) one or several microprocessor or microcontroller or digital signal processor or single purpose processors (d) not have single purpose processor (s).

7. An embedded system has RAM memory [a]
(a) for storing the variables during program run, stack and input or output buffers, for example, for speech or image (b) for storing all the instructions and data (c) for storing the programs from external secondary memory (d) for fetching instructions and data into cache(s).

8. An embedded system is [d]
[a] An Electronic system [b] A pure mechanical system
[c] An electro-mechanical system [d] either [a] or [c]

9. Which of the following is not true about embedded system? [b]
[a] Built around specialized hardware [b] Always contain an operating system
[c] Execution behavior may be deterministic [d] All the above [e] none of these

10. Which of the following is not an example of a 'Small scale embedded System'?[a]
[a] Electronic Barbie doll [b] simple calculator

[c] Cell phone [d] Electronic toy car

11. The first recognized modern embedded system is [b]
[a] Apple computer [b] Apollo Guidance Computer (AGC)
[c] Calculator [d] Radio Navigation System

12. The first mass produced embedded system is [c]
[a] Minuteman-I [b] Minuteman-II
[c] Autonetics D-17 [d] Apollo Guidance system (AGC)

Unit –II: Typical Embedded Systems

Important Points / Definitions: (Minimum 15 to 20 Points covering complete topics in that unit)

- Embedded systems are domain and application specific and are built around a central core. The core of the embedded system falls into any of the following categories:
  1. General purpose and Domain Specific Processors, Microprocessors , Microcontrollers and Digital Signal Processors
  2. Application Specific Integrated Circuits. (ASIC)
  3. Programmable logic devices(PLD's)
  4. Commercial off-the-shelf components (COTs)
- Almost 80% of the embedded systems are processor/ controller based.
- A microprocessor is a dependent unit and it requires the combination of other hardware like memory, timer unit, and *interrupt* controller, etc. for proper functioning.
- Architectures used for processor design are Harvard or Von- Neumann.
- RISC and CISC are the two common Instruction Set Architectures (ISA) available for processor design.
- Some embedded system application require only 8 bit controllers whereas some requiring superior performance and computational needs demand 16/32 bit controllers.
- DSP are powerful special purpose 8/16/32 bit microprocessor designed to meet the computational demands and power constraints of today's embedded audio, video and communication applications.
- DSP are 2 to 3 times faster than general purpose microprocessors in signal processing applications. This is because of the architectural difference between DSP and general purpose microprocessors.
- ASICs is a microchip design to perform a specific and unique applications.
- Because of using single chip for integrates several functions there by reduces the system development cost.
- As a single chip ASIC consumes a very small area in the total system. Thereby helps in the design of smaller system with high capabilities or functionalities.
- FPGA offers highest amount of performance as well as highest logic density, the most features. system is designed to perform a specific task whereas as per definition the general purpose computer is meant for general use. It can be used for playing games, watching movies, creating software, work on documents or spreadsheets etc.

- A Commercial off the Shelf product is one which is used 'as- is' COTS. The COTS components itself may be develop around a general purpose or domain specific processor or an ASICs or a PLDs.
- The major advantage of using COTS is that they are readily available in the market, are chip and a developer can cut down his/her development time to a great extent
- **Sensor:** A Sensor is used for taking Input , It is a transducer that converts energy from one form to another for any measurement or control purpose
  Ex. A Temperature sensor
- **Actuator:** Actuator is used for output. It is a transducer that may be either mechanical or electrical which converts signals to corresponding physical actions.
  Ex. LED (Light Emitting Diode)
- Common examples of onboard interfaces are:
  - Inter Integrated Circuit (I2C)
  - Serial Peripheral Interface (SPI)
  - Universal Asynchronous Receiver Transmitter (UART)
  - 1-Wire Interface
  - Parallel Interface
- **External or Peripheral Communication Interfaces**
  These are used for external communication of the embedded system i.e: communication of different components present on the system with external or peripheral components/devices.
  Common examples of external interfaces are:
  - RS-232 C & RS-485
  - Universal Serial Bus (USB)
  - IEEE 1394 (Firewire)
  - Infrared (IrDA)
  - Bluetooth
  - Wi-Fi
  - Zig Bee
  - General Packet Radio Service (GPRS)

  Example: RS-232 C & RS-485

Questions (Minimum 6 to 8)

7. What do you mean by core of the embedded system? What is its significance? What are the possible options that can be used as a core?
8. Distinguish between Microprocessor & Microcontroller
9. Explain the different types of processors according to their system bus architecture
10. Explain the different types of processors according to Instruction set Architecture
11. Explain the different types of processors according to Endianness
12. Write short note on :
    i. DSP
    ii. PLD
    iii. ASIC
    iv. COTS
13. Explain Communication Interfaces with respect to embedded system

14. Explain the following with example:
    1. Onboard communication interface
    2. Peripheral communication interface
15. Find out information and write case studies on the following communication interfaces:
    i. Infrared
    ii. WiFi
    iii. Zigbee
    iv. UART

Fill in the blanks / choose the Best: (Minimum 10 to 15 with Answers)

13. Which of the following is (are) an intended purpose(s) of embedded systems? [d]
[a] Data Collection [b] Data Processing [c] Data Communication
[d] all of these [e] none of these

14. Which of the following is an (are) example(s) of embedded system for data communication? [b]
[a] USB Mass storage device [b]Network router [c] Digital camera [d]Music player [e]all the above [f] none of these

15. A digital multi meter is an example of an embedded system for [b]
[a] data communication [b] Monitoring [c] Control
[d] All of these [e] none of these

16. Which of the following is an (are) example(s) of an embedded system for signal processing? [a]
[a] Apple iPOD (media player device) [b] SanDisk USB mass storage device
[c] Both a and b [d] none of these.

17. Embedded hardware/software systems are basically designed to [d]
[a] Regulate a physical variable [b] Change the state of some device
[c] Measure/Read the state of a variable/device [d] Any/all of these

18. Little Endian Processor means [a]
[a] Store the lower order byte of the data at the lowest address and the higher order byte of the data at the highest address memory
[b] Store the higher-order byte of the data at the lowest address and the lower order-byte of the data at the highest address of memory
[c] Store both higher order and lower order byte of the data at the same address of memory
[d] Store both higher order and lower byte of the data at the higher address of memory
[e] Store both higher order and lower byte of the data at the lower address of memory

19. An integer variable with value 255 is stored in memory location at 0x8000. The processor word length is 8 bits and the processor is big endian processor. The size of integer is

considered as 4 bytes in the system. What is the value held by the memory location 0x8000? [b]

[a] 0xFF [b]0x00 [c]0x01 [d]255 [e]256 [f]none

20. The instruction set of RISC processor is [a]

[a]Simple and lesser in number [b] complex and lesser in number

[c] Simple and larger in number [d] Complex and larger in number

21. Which of the following is true about CISC processor? [c]

[a] The instruction set is non-orthogonal[b]The number of general purpose registers is limited. [c] Instructions are like macros in C language [d] Variable length instructions

[e] all the above [f]None of the above

22. Which of the following processor architecture supports easier instruction pipelining? [b]

[a]Harvard [b] Von Neumann [c]Both of them [d] None of these

23. Microprocessors/controllers based on the Harvard architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses. State true or false [a]

[a] true [b] false

24. Embedded systems are application and domain specific. State true or false [a]

[a] True [b] False

Unit –III: Embedded Firmware

Important Points / Definitions: (Minimum 15 to 20 Points covering complete topics in that unit)

- The code to be run on the target embedded system is always developed on the host computer. This code is called the *binary executable image* or simply *hex code*.
- The process of putting this code in the memory chip of the target-embedded system is called Downloading.
- There are two ways of downloading the binary image on the embedded system: **Using a Device Programmer, and using In System Programmer (ISP).**
- An Emulator allows you to examine the state of the processor on which that program is actually running. It is itself an embedded system, with its own copy of the target processor, RAM, ROM, and its own embedded software
- A simulator is a completely host-based program that simulates the functionality and instructions set of the target processor.
- Watchdog Timer is hardware equipment. Special purpose hardware protects the system from software hangs. It always counts down from some large number to zero to resets the embedded processor and to restarts the software.

- A power-on **reset** (PoR) is a **circuit** that provides a predictable, regulated voltage to a microprocessor or microcontroller with the initial application of power. A PoR system can be a peripheral, but in sophisticated processors or controllers the PoR is integrated on the main chip.
- A **brownout reset** is a circuit that causes a computer processor to **reset** (or reboot) in the event of a **brownout**, which is a significant drop in the power supply output voltage. **Brownouts** can occur for brief intervals or over extended periods.
- **Firmware** is **programming** that is written to a hardware device's non-volatile memory. **Firmware**, which is added at the time of manufacturing, is used to run user programs on the device and can be thought of as the software that allows hardware to run.
- Writing code for embedded systems is not the same as writing the user code for a PC (personal computer). It is more like writing a driver for a PC. If no special embedded operation system (OS) like TinyOS is used the firmware engineer has to take care of all basic things like setting up hardware registers himself. The code should be minimalistic, efficient, real-time, stable, easy to read etc.

Questions (Minimum 6 to 8)

16. Explain the process of Downloading embedded software code
17. Explain the operation of Real time clock.
18. Explain the working of Brown out circuit.
19. How watchdog timer helps the circuit in protection?
20. Explain the difference between software and firmware.
21. Give examples of different firmware's.

Fill in the blanks / choose the Best: (Minimum 10 to 15 with Answers)

25. Which of the following are header files? [a]
a) #include
b) file
c) struct()
d) proc()

26. Which is the standard C compiler used for the UNIX systems? [c]
a) simulator
b) compiler
c) cc
d) sc

27. Which compiling option is used to compile programs to form part of a library?[a]
a) -c
b) -p
c) -f
d) -g

28. Which compiling option can be used for finding which part of the program are consuming most of the processing time? [c]

a) -f
b) -g
c) -p
d) -c

29. Which compiling option can generate symbolic debug information for debuggers?[d]
a) -c
b) -p
c) -f
d) -g

30. Which of the following is also known as loader? [b]
a) locater
b) linker
c) assembler
d) compiler

31. Which of the following gives the final control to the programmer? [a]
a) linker
b) compiler
c) locater
d) simulator

32. Which command takes the object file and searches library files to find the routine calls? [d]
a) simulator
b) emulator
c) debugger
d) linker

Unit –VI: RTOS based Embedded System Design

Important Points / Definitions: (Minimum 15 to 20 Points covering complete topics in that unit)

- An existing commercial OS can be used for an embedded system by adding:
  - real time capability
  - streamlining operation
  - adding necessary functionality
- Typical characteristics of RTOS include:
  - fast and lightweight process or thread switch
  - scheduling policy is real time and dispatcher module is part of scheduler
  - small size
  - responds to external interrupts quickly
  - minimizes intervals during which interrupts are disabled
  - provides fixed or variable-sized partitions for memory management
  - provides special sequential files that can accumulate data at a fast rate
- Examples of RTOS are: µCos , TinyOS , Winc, Symboin, Android etc.
- Reasons for unfit of ***desktop OS in embedded systems are :***

- Monolithic kernel is too feature reach.
- Monolithic kernel is not modular, fault-tolerant, configurable,modifiable,etc.
- Takes too much memory space.
- It is often too ressource hungry in terms of computation time.
- Not designed for mission-critical applications.
- Timing uncertainty too large.

- A real-time operating system is an operating system that supports the construction of real-time systems.
- Three key requirements:
  - The timing behavior of the OS must be predictable.
  - OS must manage the timing and scheduling.
  - The OS must be fast.
- Hard Real Time System is the one that leads to Failure to meet deadlines is fatal
  example : Flight Control System
- Soft Real Time System are late completion of jobs is undesirable but not fatal. System performance degrades as more & more jobs miss deadlines
  Ex:Online Databases
- The monolithic kernel manages the system resources between application and hardware of the system. The **user services** and **kernel services** are implemented under **same address space.** This increases the **size of the kernel further increases** the size of operating system.
- Microkernel being a kernel manages all system resources. But in a microkernel, the **user services** and the **kernel services** are implemented in **different address space**. The user services are kept in **user address space**, and kernel services are kept under **kernel address space**.
  This **reduces** the **size of the kernel** and further reduces the size of operating system.
- Task: A task or job is an executable program and it is an independent process. No task can call another task. The OS can only block a running task. It runs when it is scheduled to run by the OS(Kernel), which gives the control of the CPU on a task request(system call) or a message.
  Ex: one word file at a time
- Process: It is an instance of a computer program that is being executed. Multiple instances of the same program can execute simultaneously. A process needs various system resources like CPU for executing the process, memory for storing the code corresponding to it and associated variables, I/O devices for information exchange. It is sequential in execution.
  Ex: Writing of more than one word file, or inserting a picture/shape in word file(word file becomes main thread, and picture/shape becomes second thread)
- A thread is the primitive that can execute code. A thread is a single sequential flow of control within a process. A Process can have many threads of execution. Different threads, which are part of a process, share the same address space, i.e. they share the data memory, code memory and heap memory area. Threads maintain their own thread status i.e. CPU register values, PC and stack.
  Ex: Automatic spell check and automatic save of a file while writing a word file.
- Multi Processing: The ability to execute multiple processes simultaneously.
- Systems, which are capable of performing multiprocessing, are known as multiprocessors systems.

- Multiprocessor systems possess multiple CPUs and can execute multiple processes simultaneously.

Questions (Minimum 6 to 8)

22. What are the differences between general purpose OS and Embedded OS?
23. What are the differences between task, thread and process?
24. Identify standard Threads based in UCOS OS?
25. Write the features of Task.
26. What is semaphore? Explain with examples.
27. Explain about Context with neat diagram.
28. Write about types of multi tasking.

Fill in the blanks / choose the Best: (Minimum 10 to 15 with Answers)

33. What is the output of the specific code?

Char str1[]="Hello"

Char str2[]="World";

Str1=str2;

Printf("%s\n", str1);

A: Illegal, str1=str2 is not valid for string copy. correct operation is strcpy(str1,str2);

34. What is the output of the specific code?

Char str1[]="Hello"

Char str2[]="World";

Str1+=str2;

Printf("%s\n", str1);

A: Illegal, str1+=str2 leads to str1=str1+str2, addition of strings is not valid. Or addition of string with a constant "Hello" string is not valid. Correct operation is strcat(str1, str2).

35. Translation of assembly code to machine code is performed by---------

A. Assembler

36. What is ORG in the following piece of code

Org 4000h

SJMP 6000h

A: pseudo code

37. Give the name of file inclusion preprocessor directive?

A: #include

38. The user application and kernel interface is provided through--------

A: System call

39. The memory area which holds the program code corresponding to the core OS applications/services is known as -------

A: (a) user space

40. Which is an example of RTOS

A: (e) both a and d -> Windows CE, & QNX

41. Missing any deadline leads to catastrophic error is true for -------systems

A: hard real time systems

42. Say true or false: A Process is sequential in execution.

A: True

43. A Process has

(a) stack memory (b) program memory (c) working registers (d)data memory (e) all of these

A: all of these (e)

44. Say True or False :The stack memory of a process holds all temporary data such as variables local to the process.

A: True

45. The data memory of process holds

A: global variables

46. A Process when loaded to the memory is allocated a virtual memory space in the range 0x08000 to 0x08FF8. What is the content of the stack pointer of the process when it is created?

A: 0x7FFF

47. The state at which a process is being created is referred as ----------

A: Created state

48. The state, where a process is incepted into the memory and awaiting the processor time for execution, is known as -------

A: Ready state

49. Say True or False : Thread is also known as light weight process.

A: True

50. Different threads, which are part of a process, share the same address space. State True or False.

A: False

51. The CPU allocation for a process may change when it changes its state from

(a) 'running' to 'ready' (b) 'ready' to 'running' (c) 'running' to 'blocked'

(d) 'running' to 'completed' (e) b and c (f)a and d

A: (e) b and c

52. Multi Processor System contain

A: (b) Multiple CPUs

Unit –V: Task Communication

Important Points / Definitions: (Minimum 15 to 20 Points covering complete topics in that unit)

- Types of multi tasking are as follows:
  - **Co-operative Multi Tasking**: Any task/process can **hold the CPU as much time as it wants**.
  - **Preemptive Multi Tasking**: The currently running **task/process is preempted** to give a chance to other tasks/process to execute. The preemption of task is **based on time slots or task/process priority .**
  - **Non Preemptive Multi Tasking:** The task will be **allowed to execute until it terminates** ie. Completed state **or enters to Block/wait state**, waiting for an I/O or system resource.
- Determining which task/process is to be executed at a given point of time is known as **task/process scheduling**.
- The **kernel service/application**, which implements the scheduling algorithm, is **known as 'scheduler**'.
- Factors behind scheduling are :
  - **CPU Utilization**: How much time the CPU is busy in execution.
  - **Throughput**: No. of Processes executed per unit time.
  - **Turnaround Time**: Amount of time taken for completion of process execution.
  - **Waiting Time**: Amount of time spent by a process in the ready queue to get the CPU time for execution.
  - **Response Time**: time elapsed between the submission of process and its first response.
  - **A good scheduling algorithm has high CPU utilization, minimum turn around time (TAT), max throughput and least response time**.
- Types of Task scheduling re
  - Clock Driven Scheduling

- Preemptive scheduling:
- Preemptive SJF scheduling/Shortest Remaining Time (SRT)
- Round Robin (RR)/ Weighted Round Robin Scheduling
- Non Preemptive scheduling:
- First come First served (FCFS)/FIFO Scheduling
- Last come First served (LCFS)/LIFO Scheduling
- Shortest Job First (SJF) Scheduling
- Priority based scheduling

- A reentrant function can be **used by more than one task** without fear of data corruption, It **can be invoked while already in the process** of executing. It can be interrupted and resumed at any time without loss of data. It uses local variables (CPU registers or variables on the stack). Protect data when global variables are used.
- A semaphore is a key that your code acquires in order to continue execution. If the key is already in use, the requesting task is suspended until the key is released. There are two types
- Binary semaphores: 0 or 1 and Counting semaphores >= 0
- A counting semaphore is used when a resource can be used by more than one task at the same time
  Example: Managing a buffer pool of 10 buffers
- Types of Remote Procedure /call are :
  - Synchronous RPC: The process which calls the remote procedure is blocked until it receive a response back from the other process.
  - Asynchronous RPC: The calling process continues if execution while the remote process performs the execution of the procedure.
  - The result from the remote procedure is returned back to the caller through mechanisms like callback functions.
- The act of making processes aware of the access of shared resources by each process to avoid conflicts is known as 'Task/Process Synchronization'.
- A device driver is computer program that allows a system to interface with hardware devices.
  Example driver: printer driver, bluetooth driver, pipe driver
  Example devices: your USB stick, sensors: accelerometer
- It is a translator between the operating system and applications the use the devices and the devices. A typical operating system has many device drivers built into it. A device driver converts general IO instructions into device specific operations. Device drivers operate in a privileged mode→ requires careful design
- Selection of RTOS is based on
  - Functional requirements
  - Non Functional Requirements

Questions (Minimum 6 to 8)

29. Explain about Inter task Communication and Synchronization with example.
30. Write about Shared Memory Concepts.
31. What are the Characteristics of shared memory systems.
32. Explain memory organization with respect to embedded system environment.
33. Explain about Caches and Cache Coherence.
34. How to avoid deadlock?

35. How to choose RTOS?
36. Explain about remote procedure call with examples.

Fill in the blanks / choose the Best: (Minimum 10 to 15 with Answers)

53. Expand the following

1. **SJF : shortest job first**
2. **LCFS: last come first serve**
3. **SRT: shortest remaining time**

54. The binary semaphore implementation for exclusive resource access under certain OS kernel is named as-----

A: Mutex

55. The condition in which a process waiting for a resource held by another process which is waiting for a resource held by the first process is known as-------

A: Deadlock

56. The act of preventing the access of a shared resources by a task/process when it is being held by another task/process is --------

A: Mutual Exclusion

57. Name few standard Threads

A: Posix, Win32

58. Pick functional requirements from the following:

Cost, Real time capabilities, Processor support, Ease of use, Development language support

A: Real time capabilities, Processor support, Development language support

59. Average waiting time from Fig 1 is **----------**

A: (0+10+15)/3=> =25/3=> =8.33ms

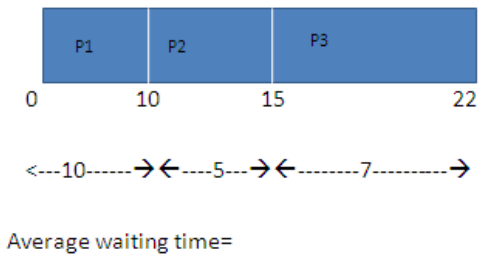

Fig.1

60. Average execution time from Fig.2 is ----------

A: (10+5+7)/3=> =22/3=> =7.33ms

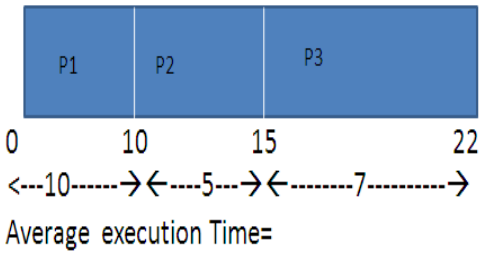

Fig.2

61.Turn Around Time TAT for P1 from Fig.3 is --------

A: 10ms

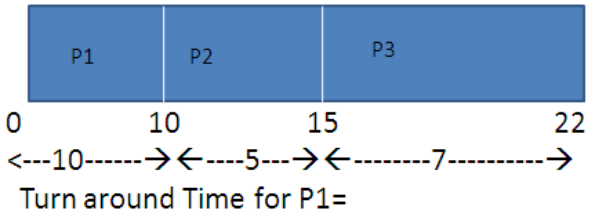

Fig.3.

62. Waiting time for P3 from Fig.4 is ---------

A: 15ms

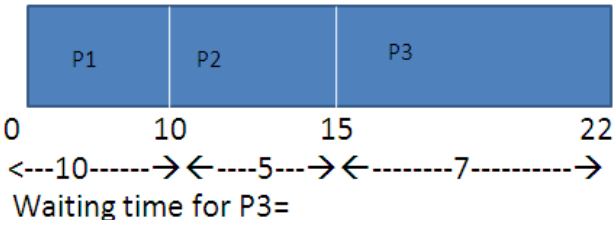

Fig.4.