Record and Reply Messages In Real Time Systems

Eman Yousif Nasir

Received on:2/9/2009
Accepted on:5/11/2009

Abstract

Many applications with real-time demands are composed of mixed sets of tasks with a variety of requirements. These can be used in the form of standard timing constraints or flow of data. Therefore you need real time multitasking operating system to guarantee within a fixed amount of time. Scheduling Algorithm for real-time multitask operating system is implemented according to the priority of tasks depending on nature of their jobs and need to resource or time. The Scheduling algorithm which used here is event driven, not clock driven.

It is necessary in data communication system which is an example of real time system to record the received messages for documentation and for further use like for simulation or computation or reply them. The time which saved with each message is important to be close to the real situation. The system is written in C++ programming language under DOS environment.

تسجيل وإعادة بث الرسائل في أنظمة الوقت الحقيقي

الخلاصة

يستخدم نظام الوقت الحقيقي عندما تتطلب أهمية الوقت بشدة في عمليات المعالج الدقيق أو في تدفق البيانات. لذلك وجدت الحاجة لاستخدام نظام التشغيل متعدد المهام والذي يعمل في الوقت الحقيقي لضمان العمل بالوقت المحدد المطلوب. ولقد تمت برمجة برنامج حدولة لنظام تشغيل متعدد المهام يعمل حسب أسبقية المهام بالاعتماد على طبيعة عملها وحاجتها للمصادر الخارجية وللوقت.

هناك حاجة ماسة في أنظمة تناقل البيانات والتي هي مثال لأنظمة الوقت الحقيقي، لخزن الرسائل المستلمة للتوثيق وللاستخدامات المستقبلية كعمل نظام محاكاة باستخدام هذه الرسائل المخزونة أو لاجراء الحسابات اللاحقة أو اعادة بثها من جديد. لذلك فان خزن الوقت مع كل رسالة ضروري للوصول الى الوضع الحقيقي للحدث الذي جرى مسبقا. تمت برمجة النظام باسخدام لغة C++.

*** Laser and Optoelectronics Engineering Department, University of Technology/Baghdad**

https://doi.org/10.30684/etj.28.3.16
2412-0758/University of Technology-Iraq, Baghdad, Iraq

Introduction

A real-time system is used when there are rigid time requirements on operation of a processor or the flow of data, and thus is often used as a control device in a dedicated application. Sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to modify the sensor inputs. Systems that control scientific experiments, medical imaging systems, industrial control systems, and some display systems are real-time systems. A real-time system has well defined, fixed time constraints. Processing must be done within the defined constraints, or the system will fail. [1]

During Record part of the system, main computer can receive messages from up to 4 remote computers which are connected to it by using separate communication link via asynchronous port (serial port), so it can receive messages simultaneously from these computers, check the validity of the messages and store them with time in files correspondent to each line, that is mean that these computers can share the main computer hard disk drive.

In the other side (During Reply part of the system), we can select segment of messages which stored in files according to the proper time to be replied, that is mean retransmit these messages to the computers or other computers connected later.

2. Data Communications

There are many situations where the user is some distance from the computer perhaps in different room or even different city or country. In such cases a data communication system may be used for transmitting data to and from a computer. [2]

A data communication system links up computer hardware and software at telecommunications facilities. It may just consist of a terminal in an office connected to a computer in a different part of the building or it may be a very complex nationwide system of interconnected computers and terminals. [2,3,4]

Well design data communication can be improved the use of computer facilities and provide fast and accurate service to everyone using data. [2]

2.1. Advantages of Data Communications:

1- Data is captured at source in computer readable form.
2- Data can be collected and information disseminated very quickly. Data is transmitted at electronics speed rather than by mail or other relatively slow methods.
3- Operating costs should be reduced.
4- There is backup data processing capacity. [2,3,4]
5- Sharing printers, plotters, modems. Etc saves money and time.
6- Increased storage capacity.

2.2. Data Security

Data Security include encrypts and decrypts data so that anyone who covertly accesses the communication channel cannot obtain confidential information, alter information as it is being transferred, or insert false messages into the stream. It authenticates the source of information. [4]

2.3. Data Compression

Data must be represented during transmission in a compact fashion in order to make optimal use of the channel. Data must be compressed at the sending end and decompressed at the receiving end. [3,4,5]

2.4. Components of Data Communications:

Any data communication system is made up of three basic elements: A sender of information, a communication link, and a receiver of information. [2]

A data communication system is made up of five basic hardware components: Computers, Communication processors (Front-end), Modems, Communication links, and Terminals. [4]

2.4.1. Communications Ports and Devices

The basic communications ports in any PC system are the serial and parallel ports. The serial ports are used primarily for devices that must communicate bidirectional with the system. Such devices include modems, mice, scanners, digitizers, and any other devices that "talk to" and receive information from the PC.[6]

(Port: That portion of a computer through which a peripheral device may communicate).

Serial Ports

The asynchronous serial interface is the primary system-to-system communications port. Asynchronous means that no synchronization or clocking signal is present, so characters may be sent with any arbitrary time spacing. [6]

Serial port is a serial communication physical interface through which information transfers in or out one bit at a time. [7] This type of communication is used over the phone system, because this system provides one wire for data in each direction. Add-on serial ports for the PC are available from many manufacturers. You usually can find these ports on one of the multifunction boards available or on a board with at least a parallel port. [6]

Serial ports may connect to a variety of devices such as modems, plotters, printers, other computers, bar code readers, scales, and device control circuits.

When a serial port is installed in a system, it must be configured to use specific I/O addresses (called ports), and interrupts (called IRQs for Interrupt ReQuest). The best plan is to follow the existing standards for how these devices should be set up.[6,7] For configuring serial ports, you should use the addresses and interrupts indicated in the Table 1.

*Note that although many serial ports can be set up to share IRQ 3 and 4 with COM1 and COM2, it is not recommended. The best recommendation is setting COM3 to IRQ 5. If ports above COM3 are required, it is recommended that you purchase a multi-port serial board. You should ensure that if you are adding more than the standard COM1 and COM2 serial ports, they use unique and non-conflicting interrupts. If you purchase a serial port adapter card and intend to use it to supply ports beyond the standard COM1 and COM2, be sure that it can use interrupts other than IRQ3 and IRQ4. [6]

2.4.2. Modems and Communication Links:

Digital data can not be directly transmitted over standard telephone lines which are designed to carry analogue signals so until the telephone system becomes fully digital, modem: modulator/Demodulator) are used, so that digital data can be sent by telephone, modulating before transmission and demodulated at the receiving end. Modems of different types handle various different transmission speeds.[2,4]

One special type of cheep low speed modem Acoustic coupler. Modems are now quite sophisticated and can perform many functions.

Data is transmitted between remote computers by mean of communication channels or networks communication traffic, depending on the distance it has to cover. [2, 4]

The physical link between computers consists of the wires or other medium that carries information from one computer to another and the interface that connects the medium to the computers.

The requirements of a link help to determine which interface to use and what medium to use to connect the nodes. The distance between computers may range from a few feet to a few thousand feet. The time between communications may be shorter than a second, or longer than a week. The number of nodes may range from two to over two hundred. [4, 8]

Most links use copper wire to connect computers, often inexpensive twisted-pair cable. The path may be a single data wire and a ground return, or a pair of wires that carry differential signals. Other options include fiber-optic cable, which encodes data as the presence or absence of light, and wireless links, which send data as electromagnetic (radio) or infrared signals in the air. [4, 8]

2.5. Communication Software

2.5.1 Programs and Processes:

The term “program” tends to mean a complete set of functions, which provide a high-level function of some sort. Therefore, in data communications, the term “process” will be used instead of “program”. Usually “process” is used when referring to some subset of functions which fit into a larger program or is part of a large system, and especially when we are talking about a program when it is executing. [2,9,10] A task is a generic term, which refers to process [1,11].

2.5.2. Protocols:

Data communications involves the transfer of data between computer programs. In order to communicate, the programs must have a common protocol. The protocol simply defines the format and meaning of the data that the programs interchange. [2, 9, 10]

A protocol is a set of rules that defines how the computers will manage their communications. The protocol may specify how data is formatted for transmitting and when and how each node may transmit. [8, 9, 10].

3. Real-Time systems

Real-time operating systems are required when the processor is involved in operations such as closed loop control and time-critical decision making. These applications require timely decisions to be made based on incoming data. For example, an I/O device samples an input signal and sends it directly to memory. Then, the processor must analyze the signal and send the appropriate response to the I/O device. In this application, the software must be involved in the loop; therefore, you need real-time multitask operating system to guarantee response within a fixed amount of time. In addition, applications requiring extended run-times or stand alone operation are often implemented with real-time operating systems. [12]

4. Multitask Operating System

Software that controls the basic, low-level hardware operations, and file management. It is provides the link between the user and the hardware. Popular operating systems include: DOS, MacOS, VMS, VM, MVS, UNIX, and OS/2. [2]

The objective of multitasking is to have some process running at all times, to maximize CPU utilization. When many processes kept simultaneously in memory, the operating system picks and begins to execute one of the processes. Eventually the process may have to wait for some event, I/O operation to complete. In a non-multitasking system, the CPU would sit idle. In a multitasking system, the operating system simply switches to and executes another process. When that process needs to wait, the CPU is switched to another process, and so on. Eventually, the first process finishes waiting and gets the CPU back. As long as there is always some process to execute, the CPU will never be idle.[1]

4.1 Classification of tasks

Many applications with real-time demands are composed of tasks of various types and constraints. Arrival patterns and importance, for example, determine whether tasks: [13]

Periodic tasks, as their name implies, execute on a regular basis; they are characterised by: (i) their period; (ii) their required execution time (per period).

The execution time may be given in terms of an average measurement and (or) a worst case execution time. [11]

Aperiodic tasks are invoked only once. Their arrival times are unknown at design time. [13] The activation of an aperiodic process is, essentially, a random event and is usually triggered by an action external to the system. Aperiodic processes also have timing constraints associated with them; i.e. having started execution they must complete within a predefined time period. Often these processes deal with critical events in the system's environment and hence their deadlines are particularly important. [11]

Sporadic tasks can arrive at the system at arbitrary points in time, but with defined minimum inter-arrival times between two consecutive invocations. [13]

Independent/Dependent tasks

Given a real-time system, a task that is going to start execution may require to receive the information provided by another task of the system. Therefore, execution of a task should be started after finishing the execution of the other task. This is the concept of dependency. The dependent tasks use shared memory or communicate data to transfer the information generated by one task and required by the other one. While we decide about scheduling of a real-time system containing some dependent tasks, we should consider the order of the starting and finishing time of the tasks [14]

Tasks whose progress is not dependent upon the progress of other processes are termed independent. While Interdependent processes can interact in many ways including communication and precedence relationships. [11]

Preemptive/Non-preemptive tasks

In some real-time scheduling algorithms, a task can be preempted if another task of higher priority becomes ready. In contrast, the execution of a non-preemptive task should be completed without interruption, once it is started [14]

Fixed/Dynamic priority tasks

In priority driven scheduling, a priority is assigned to each task. Assigning the priorities can be done statically or dynamically while the system is running [14]

4.2 Definitions of properties of multitasks operating system

Each task occurring in a real-time system has some timing properties. These timing properties should be considered when scheduling tasks on a real-time system. The timing properties of a given task refer to the following items [14]:

Release time (or ready time): Time at which the task is ready for processing.

Deadline: Time by which execution of the task should be completed, after the task is released.

Minimum delay: Minimum amount of time that must elapse before the execution of the task is started, after the task is released.

Maximum delay: Maximum permitted amount of time that elapses before the execution of the task is started, after the task is released.

Worst case execution time: Maximum time taken to complete the task, after the task is released. The worst case execution time is also referred to as the worst case response time.

Run time: Time taken without interruption to complete the task, after the task is released.

Weight (or priority): Relative urgency of the task.

For a given set of tasks, the general scheduling problem asks for an order according to which the tasks are to be executed such that various constraints are satisfied. Typically, a task is characterized by its execution time, ready time, deadline, and resource requirements. The execution of a task may or may not be interrupted (preemptive or non-preemptive scheduling). Over the set of tasks, there is a precedence relation which constrains the order of execution. Specially, the execution of a task cannot begin until the execution of all its predecessors (according to the precedence relation) is completed. The system on which the tasks are to be executed is characterized by the amounts of resources available [14].

4.3 File Management

File management is one of the most visible components of an operating system. Computers can store information on several different types of physical media. Magnetic tape, magnetic disk, and optical disk are the most common media. Each of these media has its own characteristics and physical organization. Each medium is controlled by a device, such as a disk drive or tape drive, with its own unique characteristics. These properties include speed, capacity, data transfer rate, and access method (sequential or random access method).

For convenient use of the computer system, the operating system provides a uniform logical view of information storage. The operating system abstracts form the physical properties of its storage devices to define a logical storage unit, the file. The operating system maps files onto physical media, and access these files via the storage devices. [1]

A file is a collection of related information defined by its creator. Commonly, files represent programs (both source and object forms) and data. Data files may be numeric, alphabetic, or alphanumeric. Files may be free-form, such as text files, or may be formatted rigidly. A file consists of a sequence of bits, bytes, lines, or records whose meanings are defined by their creators. The concept of a file is an extremely general one. [1]

5. Messages

Message is a sequence of data bytes, which transfer between computer programs. The programs must have a common protocol in order to exchange the messages. [2]

Messages are short, ranging from a byte or two in a very simple system to hundreds of bytes in others. A computer in this type of link isn't likely to send Megabytes of data at once.

Messages may require a quick response. In some links, a message may carry emergency information and the receiving computer will need to respond quickly, either by taking direct action or by instructing another computer to handle the problem. [8]

5.1. Buffering of Messages

A link has some capacity that determines the number of messages that can reside in it temporarily. This property can be viewed as a queue of messages attached to the link. [2]

5.2. Exception Conditions

A message system is particularly useful in a distributed environment, where processes may reside at different computers. In such an environment the probability that an error will occur during communication (and processing is much larger than in a single machine environment). In a single environment, messages are usually implemented in shared memory. If a failure occurs, the entire system fails. In a distributed environment, however, messages are transferred by communication lines, and the failure of one site (or link) does not necessarily result in the failure of the entire system. [2]

5.3. Scrambled Messages

The message may be delivered to its destination, but be scrambled on the way (for example, because of noise in the communications channel). Error checking codes (such as checksums, parity and CRC) are commonly used to detect this type of error.[2]

Noise reduces the effective capacity (bandwidth) of a channel. To contend with (say) a 10% error rate, a communication system will have to add extra bits (parity bit) to the code words to compensate for bits that are lost to errors. Use the Hamming Code, which has efficiency 4/7 = 57%. [15]

6. Applied System

This system is an example of transfer messages between computers network Where Multitask operating system is used in central computer which control this network. As shown in figure 1, local terminal can be connected to the central computer through communication channel but remote computer can be connected to the central computer by using PSTN (Public Switched Telephone Network) and modems (modem: modulator / demodulator). The system is written in C++ programming language.

There is an agreed format (protocol) between the central computer and the other computers connected to it about the transmitted messages which contain header (single byte for recognition of the message), length of the message, and ending the message with check sum (single byte contain the exclusive or of all above bytes). In addition to the preparation of the message, there is another agreement about the initialization of serial ports (baud rate, byte length, parity).

At the beginning of execution the central computer's program, serial ports are initialized in addition to all other necessary initialization of data. Then special menu displayed to select the operation, which the user wants to perform (either record or reply of messages).

Central computer's program is a multitask OS program that is mean there is a scheduler which lead either **record operation** (which contain many tasks or processes correspondent to each line to receive bytes, collect them and save the whole messages with time, in addition to another task which accept orders from the user), or **reply operation** (which contain many tasks correspondent to each line to send messages in real time of interval delay between two consecutive messages, in addition to another task which accept orders from the user). This facility or technique of multitasking is very necessary to satisfy real time system and for switching between multiple tasks. As shown in flowchart 1 and 2.

Scheduling Algorithm for real-time multitask operating system is implemented according to the priority of tasks depending on nature of their jobs and need to resource or time. The Scheduling algorithm which used here is event driven, not clock driven. The tasks are co-operate together to achieve the work of the system.

The real-time scheduling algorithm used for uniprocessor non-preemptive fixed priority systems.

6.1. Record Operation

When the user select record operation, up to 4 files are created correspondent to selected line (file's name include day and start up hour and minute, the extension refers to line number) as shown in flowchart 3. In each task correspondent to each line there is a check to the occurrence of the reception of byte and if there is no byte ready to be received through this line, the task is quitted to give another task the chance to be work when the same task has the priority or chance to work again and if it is checked that there is a byte ready to be received then the byte is received and checked to see if it is the header then save this byte in buffer; quit again then continue receive the whole message and when we reach the end of the message check the validity of this message is occurred benefit from check sum byte, if the message is correct then 4 bytes include time (quantum 55ms) relative to start up time are added at the tail of the message then the whole message is saved in the file as shown in flowchart 4. We can suspend saving of all messages according to the user's order until he wants to continue saving by another order. Also he can stop the execution of program or want to close these files as shown in flowchart 5

During reception of the messages, when the midnight is happened, the opened files are closed and new files are created indicating the new day.

On the monitor, for each selected line, file's name is displayed in addition to an alarm which indicates the reception of each message.

6.2. Reply Operation

When the user select reply operation, for each selected line scrolled menu contains files' name are displayed and the user can select any one of them and he/she can see the displayed alarm which indicate whatever the file is empty or it's start and end time. The user can use ESC key to cancel the selection of file for this line, but when the user select a specific file he can choose segment of messages to be replied (some of the messages) by choose interval time of replying message within interval time of the file as shown in flowchart 6.

In each task correspondent to each line, first we check if there is no file selected to transmit through this line then this task is sleep for ever (wait without activation by another task in the system) but if the user select a file then we seek the first message to be send according to selected time and for each byte to be send we check if the port is ready to send a byte via it then the byte is send otherwise the task is quit and then continue checking on the other run. When the first message is sent then the next message is read and checked if it is within the selected time if it is o.k. then the task compute the difference of time between the time of this message and the time of the previous message and the result is become the delay value between the two consecutive messages then this task is wait for delay and give other tasks the chance to be work. When the delay of the task is over this task continue work by sending the message as byte by byte (every time check if the port is ready), after sending the whole bytes of the message another message is read and so on until reach end of file or end of selected time as shown in flowchart 7.

On the monitor, for each selected line, file's name is displayed, interval time and time progress for each message. In addition to message that indicate end of file or out of interval.

The user can speed up or slow down the sending of messages (divide the delay time between two consecutive messages by speed counter which at least equal to one), also we can stop sending message according to user's order and send messages again according to another user's order as shown in flowchart 8.

7. Conclusions and Future Work

There are many objectives that can be satisfied by using this system like:

1- Share computer hard disk drive. (Economic view and for documentation).
2- Provide security and protection for important data.
3- The main objective is that you can use these data later in simulation. This is achieved after update the main computer programs to process data (like display) in addition to reply process.

Real time multitasking operating system is necessary to guarantee within a fixed amount of time.

Notice: because the interrupt for port 3 and the interrupt for port 4 didn't work correctly, then we used the polling of the ports. If the interrupt worked correctly then the job of interrupt service routine is to receive the byte and activate the waiting task to collect the whole message and manage it.

References

[1] Abraham Silberschatz, Peter Baer Galvin. "Operating system Concepts", 5th ed., Addison-Wesely Publishing Company, 1998.

[2] Andrew S. Tanenbaum, "Computer Networks". Third edition, Prentice-Hall Inc., 1996.

[3] Kenneth Reed, "Data Network Handbook". Galgotia publications (P) Ltd., 1998.

[4] John RR Freer, "Computer Communications and Networking". Principal Consultant. Software Science Limited, 1998.

[5] Forouzan Behrouz A., "Introduction to Data Communication and Networking", McGraw-Hill Company, 1998.

[6] Interfacing the Serial / RS232 Port V5.0, http:// www.senet.com.au/~cpeacock

[7] Serial port, from Wikipedia, the free encyclopedia.

[8] Jan Axelson, "Serial Port Complete Programming and Circuits for RS-232 and RS-485 Links and Networks, ISBN 0-9650819-2-3, 1998.

[9] Torn Sheldon, "Encyclopedia of Networking", Second edition, McGraw-Hill, 1998.

[10] James E. Goldman, Philip T. Rawles, "Local Area Networks", Second edition, John Wiley & Sons Inc., 2000.

[11] *N. Audsley, A. Burns*, "REAL-TIME SYSTEM SCHEDULING", Department of Computer Science, University of York, UK.

[12] What is Real-Time?. ©National Instruments. All rights reserved. LabVIEW, National Instruments.

[13] Damir Isovi´c and Gerhard Fohler, "Efficient Scheduling of Sporadic, Aperiodic, and Periodic Tasks with Complex Constraints", Sweden.

[14] Arezou Mohammadi and Selim G. Akl, "Scheduling Algorithms for Real-Time Systems", School of Computing, Queen's

University, Kingston, Canada 2005.
[15] Peter J. Denning with Craig Martell, "Communication, Moving data reliably from one place to another", © 1998-2007

Table (1)) Standard Serial I/O Port Addresses and Interrupts

System	COMx	Port	IRQ
All	COM1	3F8h	IRQ4
All	COM2	2F8h	IRQ3
ISA bus	COM3	3E8h	IRQ4*
ISA bus	COM4	2E8h	IRQ3*

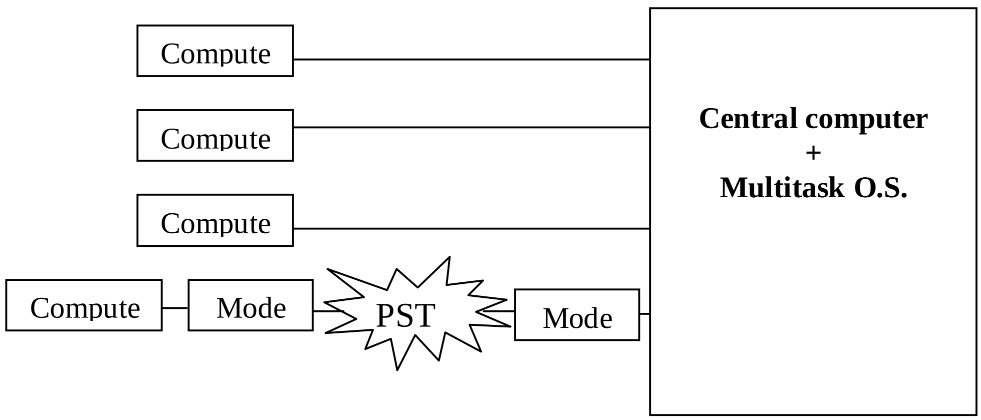
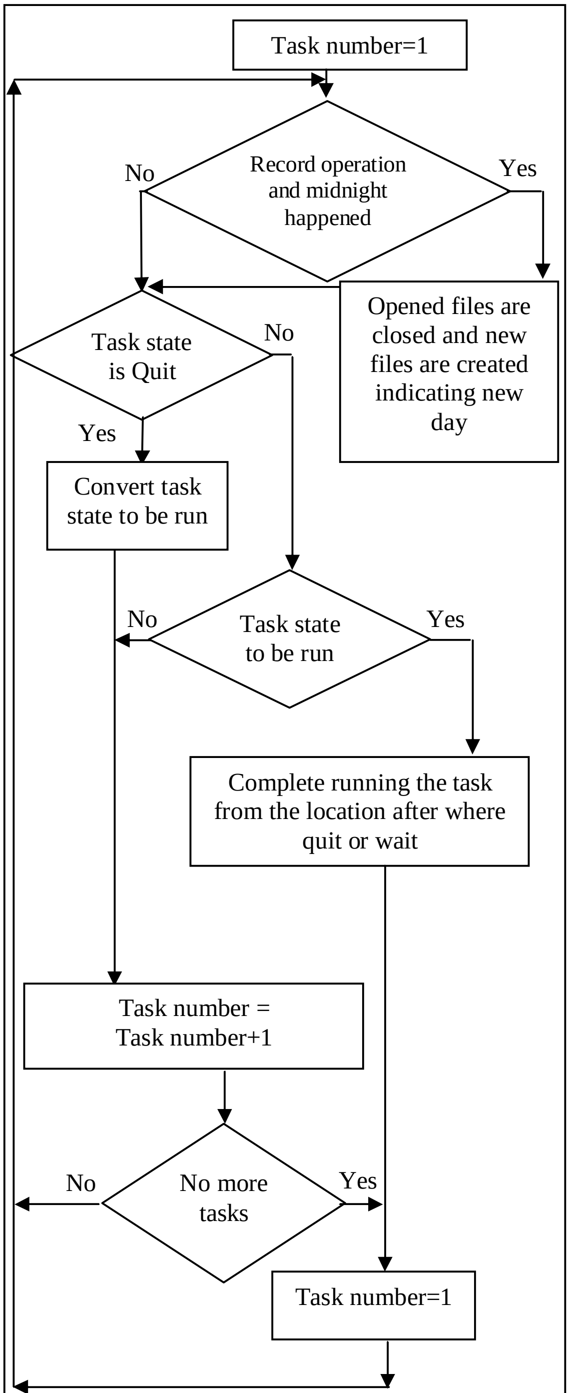
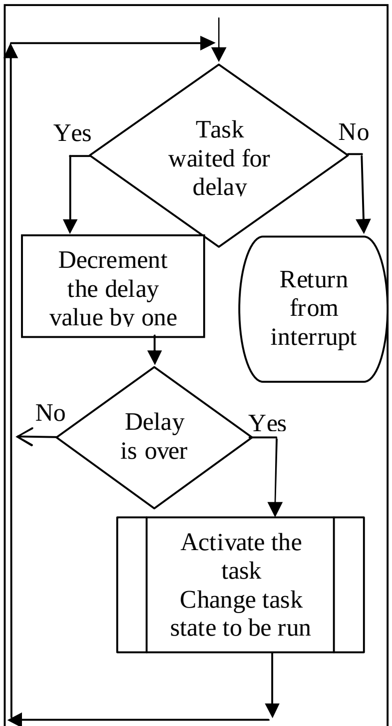


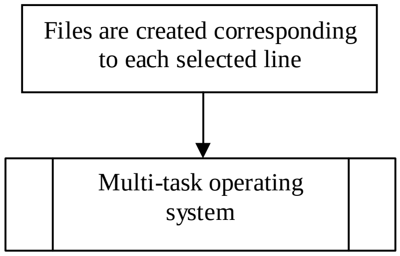
Figure (1) Computer network of record and reply system



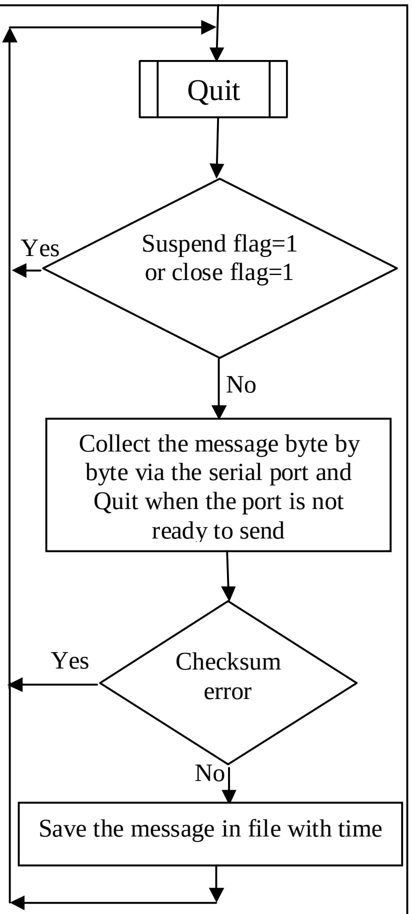
Flowchart 1: Multi-task operating system



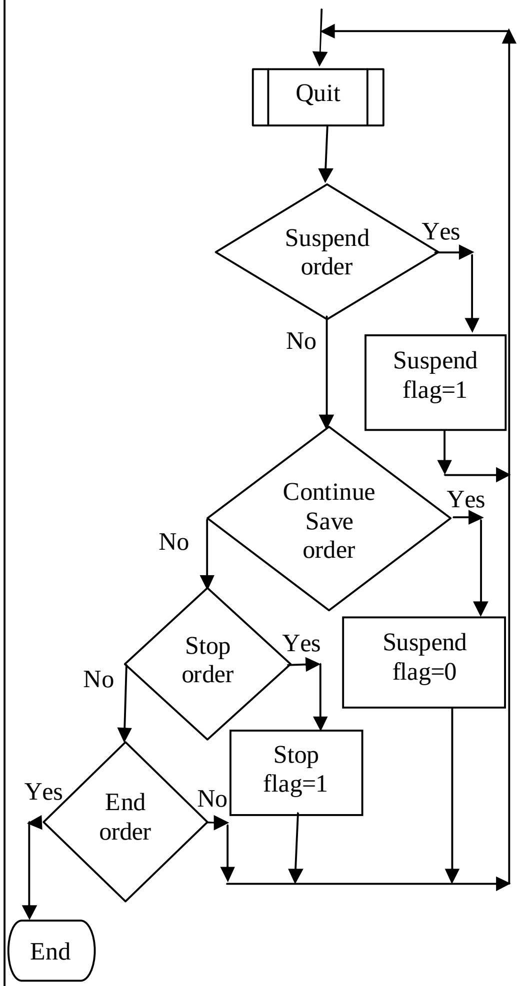
Flowchart 2: Tick time interrupt which work every 55ms



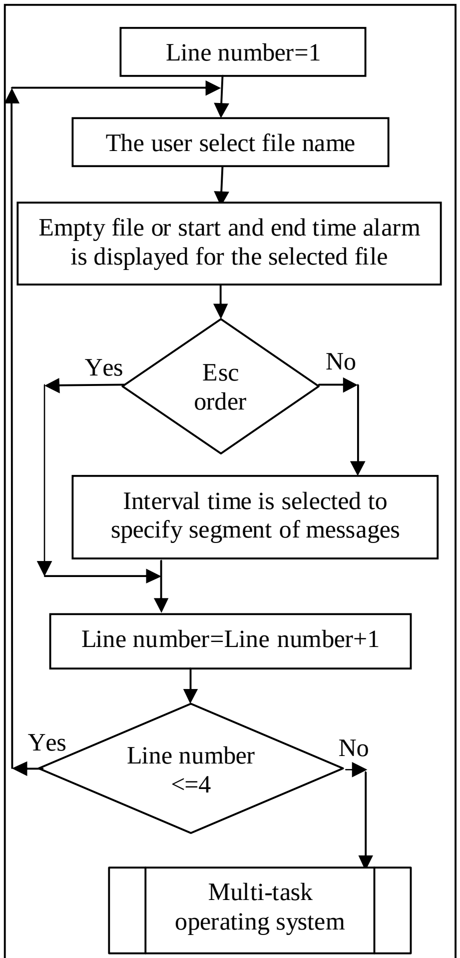
Flowchart 3: Main program of record operation



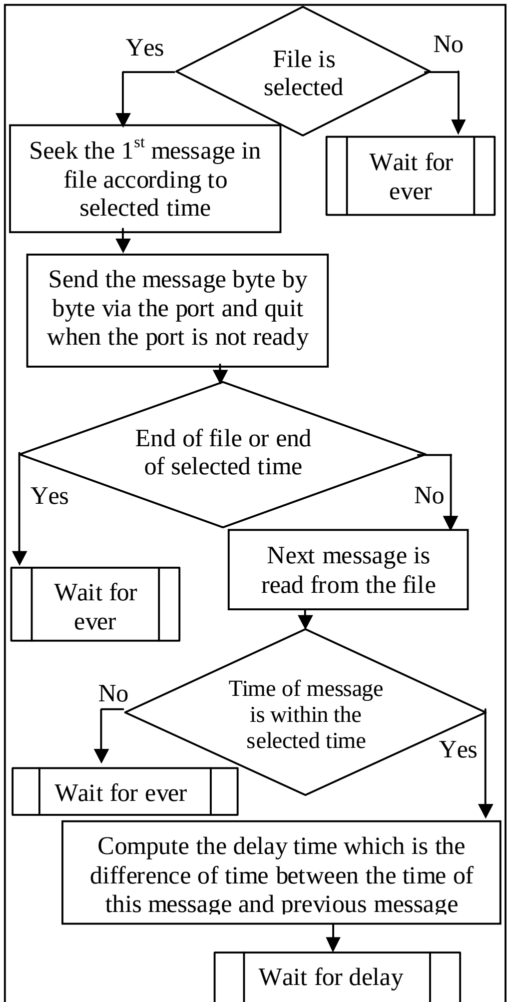
Flowchart 4: Record Task 1 up to 4



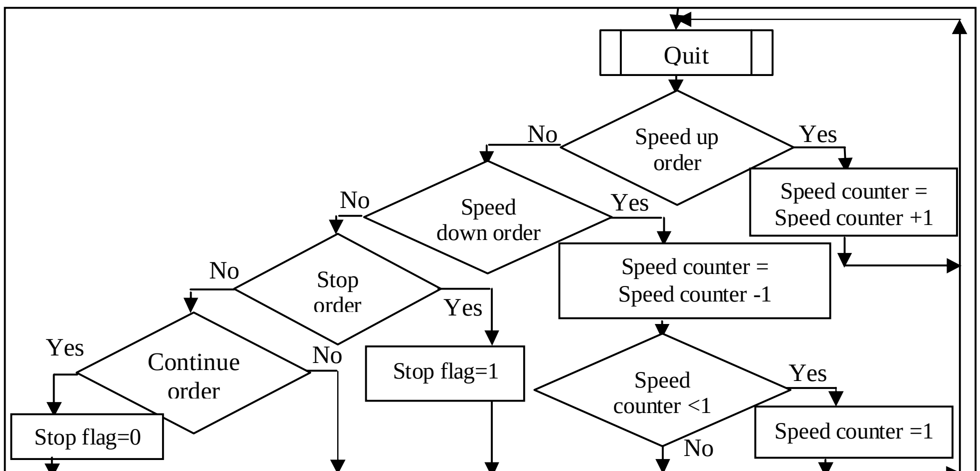
Flowchart 5: Task for user's order in record operation



Flowchart 6: Main program of reply operation



Flowchart 7: Reply Task 1 up to 4



Flowchart 8: Task for user's order in reply operation