SYSTEMATIC APPROACH TO PC MAINTENANCE & TROUBLESHOOTING

(A Case study of AMI-TECH Computers (Nig) Ltd Maintenance Workshop, Minna)

By

IDEM NKOYO INYANG.

PGD/MCS/98/99/807

A Project submitted to the Department of Mathematics/Computer Science, Federal University of Technology, Minna, Niger State, In partial fulfillment of the requirement for the award of Post-Graduate Diploma in Computer Science.

September, 2000.

CERTIFICATION

This project has been examined and found acceptable in partial fulfillment of the requirement for the post- graduate diploma in Computer Science of the department of Mathematics/Computer Science of the Federal University of Technology.

Mr.L.N. Ezeakor Project Supervisor	Date.
Dr. S. A. Reju H.O.D. Mathematics/Computer Science	Date.
External Examiner	Date.

DEDICATION

This project work is dedicated to God Almighty for His grace and mercies throughout the period of this programme.

ACKNOWLEDGEMENT

My sincere gratitude goes to the Lord Almighty for His bountiful grace upon my life and making me what I am in this life.

My profound appreciation to my project supervisor, Mr.L.N. Ezeakor for his untiring effort and understanding in making sure that this project becomes a reality.

I also wish to thank the HOD, Dr. S.A. Reju, during the course of this programme, who made sure that things were put in the right place for the success of this programme. May the Lord God reward you abundantly.

I will not forget all my lecturers in the department for their patience, courage and understanding in the course of imparting knowledge into me. I pray that the Lord God Almighty in His infinite mercies will bless you all.

My deepest gratitude to my family members, though afar off, but bear with me and gave me moral support throughout my period of absence. I still love you all.

To the Director, AMI-TECH Computers (Nig.) Ltd for his encouragement, help and support throughout the period of this programme and the entire staff of the company who help me in various ways to make this project work a success.

My special gratitude to a sister when there was none, miss Matemi Yemisi, AMI-TECH Computers (Nig.) Ltd, for her untiring effort, support and encouragement in making sure that the total resources were not wasted. I love, and appreciate your effort and I pray that the Lord in His infinite mercies will reward you more abundantly (Amen).

Finally, my regards to my friends, miss Oluwatoyin Yisa, miss Amaka Udechuckwu (friends in deed) and also to Mr. Paul Abutu, who gave me moral support and encouragement throughout the period of this project, may the Lord bless you abundantly, (Amen.)

ABSTRACT

There is considerable variation in both the quantity and quality of service information provided by Personal Computers Manufacturers. All too often technical manuals, if they can be obtained at all, are of poor quality and the service information that they contain may be extremely limited. Different problems do arise when by virtue of age; a personal computer ceases to be adequately supported by the manufacturers or by "third party" maintenance utilities. For example when the system performance is limited by the available size of the hard disk, there is need to upgrade the disk with a more modern and larger disk.

This project work therefore tends to design steps or guide to troubleshooting needs and to provide a systematic approach to system maintenance. Common faults and possible solutions are being highlighted and this is believed to aid in common troubleshooting and maintenance tasks.

TABLE OF CONTENTS

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER ONE

1.1 INTRODUCTION

Today's Pc is largely a collection of very expensive subassemblies, many of which are now manufactured in the Pacific Rim and assembled in high volumes at factories around the world.

The divers array of peripherals that are now available (e.g. tape devices, Pc cards, CD-ROMs, Pointing devices, etc) enjoy a remarkable level of hard ware compatibility using well-established interface schemes (e.g. SCSI, EIDE, USB, or PCI). The labour cost involved in a component-level repair today is usually more expensive than the cost of a replacement assembly. There is little doubt that the day of component-level PC repair is over.

However, PCs still fail in ways that continue to exhaust even the most patient mind.

Effective troubleshooting requires more than simply an arbitrary swapping of boards and drives.

Efficient and cost-effective troubleshooting requires an understanding of PC hardware and operating systems along with a keen knowledge of symptoms and diagnostic.

1.2 BACKGROUND OF AMI-TECH COMPUTER (NIG) LIMITED.

Ami-tech computers (Nig) limited was established in 1997 as a division of TRI-PEM ventures and incorporated in 1998 as a limited liability company.

The main objectives of the company are:

(1) To provide quality hardware services to Information Technology (I.T) users in Niger state and its environs.

(2) To provide an efficient training school to train computer and electronics graduates in hardware maintenance and repairs.

(3) To offer value added services to both individual and organisation within Niger and its environs.

(4) To provide an avenue to train both graduates and non-graduates in the practical application of computer.

The company started in 1998 with a total staff strength of five (5) with special emphasis to system maintenance and upgrading. Late in 1999, the company moved from Tunga (opposite CBN staff quarters) to the present Head office (Beside fire service station, Bosso Road Minna) .The Tunga office now serve as the branch office. Other branches are situated in Abuja.

The company is presently a member of ITAN (Information Technology Association of Nigeria) and has a total staff strength of fourteen (14).

1.3 OPERATION AND FUNCTIONS OF AMI-TECH COMPUTER (NIG) LIMITED

The company is committed to serving the public and in her bid to provide meaningful and comfortable services to her customers undergo the following operations.

(a)Sales of computer systems /accessories

(b) supply of computers systems

© Repairs/maintenance of computer systems

(d) Training of individuals and corporate organisation.

(e)Consultancy Services

(f) general Contractors

In order to make room for effective and efficient services, the company has several departments which take care of these operations and these include :

(i) Training department

(ii) Marketing /sales department

(iii) Maintenance department

(iv) Consultancy and

(v) General Services

Taking the topic of this project work into consideration, emphasis will be on the maintenance department.

This department is charged with the following responsibilities:

(i) Troubleshooting and repairs

(ii) Assembly / Installation of computer systems.

(iii) Maintenance of computer systems

1.3.1 TROUBLESHOOTING AND REPAIR

Troubleshooting involves measures taken to resuscitate a broken down Computer system to a working condition. The maintenance workshop of AMI-TECH computers in trying to find where there's fault may either open up the computer system (disassembly) to locate the fault (if it is a hardware problem) and fix the problem by repairing or replacing the faulty component.

Cost is usually taken into consideration in any repair operation carried out. The cost of repairing a faulty component is usually considered and the more effective option adopted. In a situation where the problem/faulty is attributed to software, necessary steps like reinstallation (if the situation demands). Fixing of errors by following and taking note of the error codes is usually taken.

1.3.2 ASSEMBLY AND INSTALLATION

The department is also committed to Assembly and Installation of computer systems. The company, builds her own systems using the components, thus all the sales and supply of new systems pass through the maintenance department.

The hardware components are build to make a system by the company and installation of operating systems and other software packages. The systems used in AMI-TECH computers are assembled by this department.

.3.3 MAINTENANCE OF THE COMPUTER SYSTEM

The department undergoes the maintenance of the company's systems and those of her customers. The systems in AMI-TECH being used for training are maintained to avoid complete breakdown. The maintenance here is that of preventive maintenance.
Covering the system after use each day prevents dust, the hard disk drive are usually reformatted every three months, defragmentation of disk when required.

DEFINITION OF BASIC TERMS

A: The identifier used for the first floppy disk drive; the operating system always checks drive A: for start-up (or boot up) instructions before checking the hard disk, drive C:.

Adapter: A printed circuit board that plugs into a computer's expansion bus to provide added capabilities. Common adapters for the PC include display adapters, memory expansion adapters, input/output adapters that provide serial, parallel and games ports, and other devices such as internal modems, CD-ROMS, or network interface cards.

Address: The precise location in memory or on disk where a piece of information is stored. Every byte in memory and every sector on a disk have their own unique addresses.

AGP: Abbreviation for Accelerated Graphic Port. AGP is a video specification that enhances and accelerates the display of three-dimensional objects.

Algorithm: A formal set of instructions that can be followed to perform a specific task, such as a mathematical formula or a set of instructions in a computer program.

Architecture: The overall design and construction of all or part of a computer, particularly the processor hardware and the size and ordering sequence of its bytes. It is also used to describe the overall design of software.

ATX: A type of computer case that rotates the processor 90 degrees on the motherboard, physically placing it and the slots in a different position. An ATX motherboard will not fit in a standard case and vice versa.

Autoexec.bat: A contraction of Automatically executed Batch. A special DOS batch file, located in the root directory of the start-up disk that runs automatically every time the computer is started or restarted.

Backup: A up-to-date copy of all files that can be used to re-load the hard disk in case of accident. It is an insurance against disk failure affecting hundreds or possibly thousands of files that might be present on the system hard disk.
Bad sector: An area on a hard disk or floppy disk that cannot be used to store data because of a manufacturing defect or accidental damage. An operating system finds, marks and isolates bad sectors.

Benchmark: A test that attempts to quantify hardware or software performance-usually in terms of speed, reliability, or accuracy.

Bios: Acronym for basic input lout put system, pronounced "bye, os," In the PC, a set of insdrictions, stored in read only memory (ROM), that let the computer's hardware and ɔperating system communicates with application programs and peripheral devices such as hard disks, printers, and video adapters.

Boot:: the loading of an operating system into memory, usually from a hard disk, although occasionally from a floppy disk or software that causes a malfunction of some sort. If the program is in software, it can be fixed by changes to the program: if the fault is in hardware, new circuits must be designed and constructed.

CHAPTER TWO

2.1 THEORY AND ANALYSIS OF SYSTEM MAINTENANCE

There is no magic prescription for fixing broken computers, though just plodding, methodical analysis to eliminate what is working right until what's left is the problem. Some involve stripping all the data and programs off the disk and starting over from scratch: others involve substituting components. There's always need for backups plus the master copies of the software to be kept handy.

Repair isn't a very pleasant subject either. Unless the problem is as simple as cleaning dust out of the machine or reseating cards and cables, there's certainly going to be swapping of components. Most problems in running computers are due to software problems

With the exception of blowing dust out of floppy disk drives and tape drives, hardware work involves taking the case of the computer apart.

These involve using some tools and this take us to the next point

2.2 SYSTEM MAINTENANCE TOOLS \ TEST EQUPMENT

A selection of basic tools and test equipment is needed before trying to tackle system repair or maintenance. Test equipment allows measurement of important circuit. Parameters such as voltage, current, resistance. capacitance and semi-conductors junction conditions.

The following tools are needed for good repair operations:

Screwdrivers. Slotted and Phillips screw divers. Screwdrivers should be the first item on the list computer assemblies are generally held together with small or medium-sized

Philips type screws Each screw should be about 4-in (10.16-cm) to 6-in (15.24-cm) long with handle for a good grip

Wrenches- is used to hold hex-shaped bolt heads or nuts.

Needle nose pliers- is valued addition to the toolbox they are used to grip or bend both mechanical and elec

Ideally, the preventive maintenance process is designed to prevent problems-not cause them. However at times, an accidental oversight or careless effort can result in problems.

Fortunately, most maintenance- related issues are easy to spot and resolve.
Below are the basic steps to take while performing preventive maintenance:

PROTECTING THE DATA

STEP1: FILE BACKUPS

By creating a 'copy" of the system files (or even just a part of them) the copy can be restored and work continues in the event of a disaster. Some backup software's are needed to format the media and handle the backup and restore operation with windows 95\98, the native backup applet is usually used (click on start, program, Accessories system tools and backup) the backup software should be compatible with one another

.

STEP 2 : CMOS BACKUPS

All PCs use a sophisticated set of configuration settings (everything from "Date" and "Time" to "memory Hole") which define how the system should be operated. These settings are stored in a small amount of very low-power memory, called CMOS RAM. Each time the PC starts, motherboard BIOS reads the CMOS RAM and copies the contents into low system memory (the BIOS Data Area, BDA), while the system power is off, CMOS contents can be lost. In most cases, this will prevent the system from even starting until you reconfigure the CMOS set up from scratch. By making a backup of the CMOS set up, the cost setting can be restored in a matter of minutes. CMOS backup Are simply printed screens of the CMOS set up page

What is needed to perform a CMOS back up is a printer. The printer should be attached to the PCs parallel port.

After starting the CMOS set up routine, visit each page of the set up and use the <***print screen*** >key to "capture" each page to the printer.

Because every Bios is written differently, be sure to check for sub-menus that might be buried under each main menu option

TIPS:

CMOS back up are quick and simple but by following the point below, the most benefits can be obtained:

- Print out every CMOS set up page
- Keep the printer pages taped to the PC's housing or with the system's original documentation.
- The CMOS set up should be back up wherever a change is made to the system contiguratiOon

CLEANING

The first set of procedures involves exterior cleaning. Items needed for cleaning include

- Mild Ammonia-based cleaner
- A supply of paper towels or clean lint-free cloths
- A canister of electronics- grade compressed air
- A small static-safe vacuum cleaner.

STEP3: CLEAN THE CASE

The case should be cleaned with cloth lightly dampened with ammonia cleaner to remove dust, dirt or stains from the exterior of the PC.

Care should be taken not to accidentally alter the CD-ROM volume or sound master controls

STEP4 CLEAN THE AIR INTAKE

The air intakes of the casing should be carefully attended to, accumulation of dust or debris around the intake should be cleaned off. If the filter is washable, it might be rinsed in simple soap solution and water for best cleaning. If there's no intake fitter, simply clean around the intake area.

STEP 5: CLEAN THE SPEAKERS

Multi Media speaker offer a countless number of ridges and openings there are just perfect for accumulating dust and debris.

Compressed air can should be used to gently. dust out speaker's opening. The long-thin air nozzle should not be inserted into speaker because it can easily puncture the speaker core and ruin it but the nozzle should be removed and air sprayed directly from the can. A clean cloth lightly dampened with ammonia solution should be used to removed dirt and stains from the speaker housing

STEP 6: CLEAN THE KEYBOARD

Accumulation can jam keys or cause repeated keystrokes. Attach long thin nozzle to the can of compressed air and use the air to blow through the horizontal gap between key rows. The keyboard should be kept way from the face as this will kick up a lot of dust. Ammonia dampened cloth should be used to remove dirt and stain from the key and keyboard housing.

If any key seems unresponsive or "sticky" the corresponding key can be removed and a bit of good quality electronic contact cleaner sprayed into the key cap.

The>ENTER>key or > should not be removed as they are held in place by metal brackets that are extremely difficult to reattach once the key is removed

STEP7: CLEAN THE MONITOR

Several issues are important when cleaning a monitor: ventilation, case and CRT.

Monitor rely no vent openings for proper cooling. The vacuum cleaner should be used to carefully remove any accumulations of dust and debris form the vent underneath the case. The vent openings should not be blocked by any of this.

NEXT is to use clean cloth, lightly the monitor's p lasts case. Under no circumstance should cleaner be sprayed directly, on the monitor housings as active circuitry is directly under the top vent.

Ammonia should not be used to clean the CRT face. The CRT is often tinted with anti-glare and other coating and even mild chemicals can react with some coatings. Clean tap water only should be used to clean CRT face and should be dried completely.

STEP8: CLEAN THE MOUSE

Mouse is partially susceptible to dust and debris which are carried from the mouse pad up into the mouse ball and rollers. When enough foreign matter has accumulated the mouse cursor hesitates or refuses to move completely.

Loosen the retaining ring and remove the mouse ball. Using a Clean cloth and and an ammonia solution. The mouse ball should be dried thoroughly and set aside with the retaining ring. Clean cloth should be used to clean the rollers completely compressed air can should be used to blow out any remaining dust or debris that might still be inside the mouse. Finally, the mouse ball should be replaced and secured into place with its retaining ring.

EXTERNAL CHECIK

The PC needs to be power up for this check. Three checks should be performed every four (4) months (three times per year)

STEP 9: CHECK THE EXTERNAL CABLES

Myriad of external cables interconnect the computers to its peripheral device each cable should be examined and verify that it is securely connected. The following cable should be checked :

- AC power cable for the PC
- AC power cable for the monitor
- AC power cable for the printer

- AC \DC power pack for the an external modem (if used)
- Keyboard cable
- Mouse cable
- Joystick cable (if used)
- Video cable form the sound board
- Serial port cable to external modem
- Parallel port cable to printer

STEP 10: CLEAN THEFLOPPY DISK DRIVE

Floppy disk are a "contact" media-the read\write head of the floppy drive actually contact the floppy disk. This contact transfers some of the magnetic oxide from the floppy disk to the drive's read /write heads.

Eventually, enough oxides can accumulate on the read \write head to cause reading or writing problems with the floppy drive. The floppy drive Should be periodically cleaned to remove excess oxides.

2.4: COMPONENTS OF A PERSONAL COMPUTER

A modern PC is both simple and complicated. It is simple in the sense that over the years many of the components used to construct a system have become integrated with other components in less actual part. It is complicated in the sense that each part in a modern system performs many more functions than did the same types in folder systems: the PC is a modular device and this modularity makes problem determination and repairs possible.

Many repairs just require finding and replacing the faulty component so the firs step is to identify what's in the "box". The components needed to assemble a basic modern PC system are listed below:-

- Motherboard
- Processor

- Memory (RAM)
- Case (chassis)
- Power supply
- Floppy drive
- Hard disk
- CD- ROM, CD-R, OR DVD-ROM drive
- Keyboard
- Mouse
- Video card
- Monitor (display)
- Sound card

MOTHERBOARD

The motherboard is the core of the system it is really the PC- everything else is connected to it and it controls in the system. Motherboards are available in several different shapes or form factors. It usually contains the following individual components:-

- Processor socket (or slot)
- Processor voltage regulators
- Chipset
- Level 2 cache (normally found in the CPU today)
- Memory (SIMM or DIMM) sockets
- Bus slots
- ROM BIOS
- Clock (CMOS battery)
- Super I/O chip

The chipset contains all the primary circuitry that makes up the motherboard; in essence, the chipset is the motherboard. The chipset controls the CPU or processor bus the L_2

cache and main memory the PCI (Peripheral component interconnect) bus, the ISA (Industry Standard Architecture) bus, system resources, and more.

If the processor represents the engine of the system the, chipset represents the chassis in which the engine is installed.

The chipset plays a big role in determining what sort of features a system can support. E.g. which processor to be used, which type and how much memory can be installed, what speed the machine can be run, and what types of system buses the system can support are all tied in to the motherboard chipset.

The ROM BIOS contains the initial POST (Power – on Self Test) Program, bootstrap loader (which loads the operating system) drives for items that are built into the board (the actual BIOS code), and usually a system set up program (often called CMOS Set up) for configuring the system.

PROCESSOR

The processor is often thought of as the engine of the computer. It is also called the CPU (Central Processing Unit), it is the single most important chip in the system because it is the primary circuit that carries out the program instructions of whatever software is being run.

Modern processors contain literally millions of transistors, etched onto a tiny square of silicon called a die, which is about the size of a thumbnail. The processor has the distinction of being one of the most expensive parts of most computers, even though it is also one of the smallest parts.

Intel is generally credited with creating the first microprocessor in 1971 with the introduction of a chip called the 4004. Today, Intel have almost total control over the processor market, at least for PC systems, this means that all PC-compatible systems use either Intel Processor or Intel compatible processors from a handful of competitors (such as AMD or Cyrix).

In some ways the success of the PC, and the Intel architecture it contains has limited the growth of the PC. In other ways, however, its success has caused a huge number of programs, peripherals, and accessories to be developed, and the PC to become a de facto standard in the industry. The original 8088 processor used in the first PC contained closed to 30,000 transistors and ran at less than 5mhz.
The most recent processors from Intel contain close to 30million transistors and run at over 500mhz. Intel has already demonstrated processors running at 1Ghz.

Processors can be identified by two main parameters: how wide they are and how fast they are. The speed of a processor is a fairly simple concept. Speed is counted in megahertz (MHz), which means millions of cycles per second -and faster is better !. The width of a processor is a little more complicated to discuss because there are 3 main specifications in a processor that are expressed in width. They are:

- Data input and output bus
- Internal registers
- Memory address bus.

MEMORY (RAM)

The system memory is often called RAM (Random Access Memory). This is the primary memory, which holds all the programs and data the processor is using at a given time.
RAM requires power to maintain storage, so when the computer turn off everything in the RAM is clear; when it is turn back on, the memory must be reloaded with programs for the processor to run. The initial program for the processor come from a special type of memory called ROM (Read only memory), which is not erased when the power to the system is turned off. The Rom contains instructions to get the system to load or boot an operating system and other programs from one of the disk drives into the main RAM memory so that the system can run normally and perform useful work.
Newer operating systems allow several programs to run at one time, with each program or data file that is loaded using some of the main memory
Generally, the more memory the system has, the more programs you can run simultaneously. Memory is the workspace for the computer processor. It is a temporary

storage area where the programs and data being operated on by the processor must reside. Memory storage is considered temporary because the data and programs remain there only as long as the computer has electrical power or is not reset. Before being shut down or reset, any data that has been changed should be saved to a more permanent storage devices of some type (usually a hard disk) so it can be reloaded into memory again in the future.

Main memory is called Ram because it can randomly (and quickly) access any location in memory. Rom is also randomly accessible, yet is normally differentiated from the system RAM because it cannot normally be written to. RAM can refer to both the physical chips that make up the memory in the system and the logical mapping and layout of that memory. Logical mapping and layout refer to how the memory addresses are mapped to actual to actual chips and what address locations contain which types of system information. Memory temporarily stores programs when they are running, along with the data being used by those programs. RAM chips are sometimes termed volatile storage because when the computer is turn off or an electrical outage occurs, whatever is stored in RAM is lost unless it is saved to the hard drive. Physically, the main memory is a collection of chips or modules containing chips that are normally plugged into the motherboard. These chips or modules vary in their electrical and physical design and must be compatible with the system into which they are being installed to function properly.

CASE (CHASSIS)

The case is the frame or chassis that houses the motherboard, power supply, disk drives, adapter cards, and any other physical components in the system. There are several different styles of cases available from small or slim versions that sit horizontally on a desktop to huge tower types that stand vertically on the floor. In addition to the physical styles, different cases are designed to accept different form factor motherboards and power supplies.

Some cases have factor that make installing or removing components easy, such as a screw less design that require no tools to disassemble, side open panels or trays that allow easy mother board access, removable cages or brackets that give easy access to disk drive, e.t.c some cases include additional cooling fans for heavy duty systems, and some are even available with air filters that ensure that the interior will remain clean and dust free. Most cases include power supply but power supplies and cases can also be purchase separately.

FLOPPY DISK DRIVE

The floppy drive is a simple, inexpensive. low capacity removable media magnetic storage device. FLOPPY disks were the primary medium for software distribution and system backup for yeas with the advent of CD, ROM and DVD-ROM discs as the primary method of installing or loading new software in a system, and with inexpensive high capacity tape dives for back up the floppy dive is not used very often in most modern systems. Floppy drive is the first device from which a PC attempts to boot, and it is still the primary method that is used for loading initial operating system start up software and core hardware diagnostics.

Recent advancement in technology have created new types of floppy drives with more than 120MB storage, making the drive much more usable for temporary backups or for moving files from system to system.

HARD DISK DRIVE

A hard disk drive is a sealed unit that a PC uses for non-volatile data storage. Non-volatile or permanent storage device retains the data even when there is no power supplied to the computer.

Because the hard drive is expected to retain its data until a user deliberately erase it, the PC uses it to store its most crucial programming and data. As a result, when the hard disk fails, the consequences are usually very serious.

An understanding of the hard disk unit function is needed to maintain service and expand a PC system properly.

A hard disk drive contains rigid, disk-shaped platters usually constructed of aluminium or glass. Unlike floppy disks, the platters cannot bend or flex hence the term hard disk drive. In most hard disk drive the platters cannot be removed, the reason why they are sometimes called fixed disk drives. Removable hard disk drives are also available.

The basic physical construction of a hard disk drive consist of spinning disks with heads that move over the disks and store data in tracks and sectors. The heads read and write data in concentric rings called tracks, which are divided up into segments called sectors, which normally store 512 bytes each. Hard disk drives usually have multiple disks called platters, that are stacked on top of each other and spin in union, each with two sides on which the drive stores data.

Most drives have at least two or three platters, resulting in four or six sides and some drives have up to 11 or more platters. The identically positioned tracks on each side of every platter together make up a cylinder. A hard disk drive normally has one head per platter side, with all the heads mounted on a common carrier device or rack.

The heads move radially in and out across the disk in unison; they cannot move independently, because they are mounted on the same rack.

CD-ROM DRIVE.

The CD-ROM is a read-only optical storage medium capable of holding up to 682MB of data (approximately 333,000 pages of text). 74 minutes of high-fidelity audio or some combination of the two on a single side of a file-inch disk. The CD-ROM is similar to the familiar audio compact disc and can, in fact, play in a normal audio player. Accessing data from a CD-ROM is quite a bit faster than from a floppy disk but considerably slower than a modern hard drive.

The term CD-ROM refers to both the discs themselves and the drive that reads them. Although identical in appearance to audio CDs, CD-ROMs store data instead of (or in addition to) audio. The CD-ROM drives in PCs that read the data audio discs bear a strong resemblance to an audio CD-player. This water base is coated with a metallic film, usually an aluminium alloy. The aluminium film is the portion of the disk that the CD-ROM drive reads for information. The aluminium film or strata is then covered by a plastic polycarbonate coating that protects the underlying data. A label is usually placed on the top of the disc, and all reading occurs from the bottom. CD-ROMs are single-sided.

KEYBOARD

The keyboard is one of the basic components of the computer system. It is the primary input device and is used for entering data and commands into the system. In the years, since the introduction of the original IBM PC, IBM has created three different keyboard designs for PC systems; Microsoft has augmented

One of them. These designs have become de facto standards in the industry and are shared by virtually all PC manufacturers. With the introduction of Windows 95, a modified version of the Standard 101-Key design (Created by Microsoft) appeared, called the 104-Key Windows Keyboard.

The primary keyboard types are as follows:

- 101 - key Enhanced keyboard
- 104 - key Windows keyboard
- 83 - key PC and XT keyboard(obsolete).
- 84 - key AT keyboard (obsolete)

The Enhanced keyboard is available in several different variations, but all basically the same electrically and all can be interchanged.

The 101 - key keyboard layout can be divided into the following four sections:

- Typing Area
- Numeric Keypad

- Cursor and Screen Control
- Function Keys.

MOUSE

With the advent of computer operating systems that used a Graphical User Interface (GUI), it became necessary to have a device that enabled a user to point at or select items that were shown on the screen. Although there are many different types of pointing devices on the market today, the first and most popular device for this purpose is the mouse.

By moving the mouse across a desk or table top, a corresponding pointer can be moved across the computer screen, allowing items to be more easily selected or manipulated than they can with a keyboard alone. Stand mice, as used on PCs have two buttons, one for selecting menus, mice are also available with a third button, a wheel, or a stick, which can be used to scroll the display or for other special functions.

VIDEO CARD

The video card controls the information that is seen on the monitor. All video cards have four basic parts: a video chip, or chip set, video RAM , a DAC (Digital to Analog Converter) and a BIOS. The video chip actually controls the information on the screen by writing data to the video RAM. The DAC reads the video RAM and converts the digital data there into analog signals to drive the monitor. The BIOS holds the primary video driver that allows the display to function during boot time and at a DOS prompt in basic text mode.

More enhanced drivers are then usually loaded from disk to enable advanced video modes for windows or applications software

MONITOR (DISPLAY)

In most systems, the monitor is housed in its own protective case, separate from the system case and chassis. In portable systems and some low-cost PCs, however, the monitor is built into the system case.

Monitors are generally classified by 3 major criteria: diagonal size in inches, resolution in Pixels, and refresh rate in hertz (Hz). Desktop monitors usually range from 14" to 21" diagonal measure (although the actual viewable area is smaller than the advertised measure).

LCD monitors in portable systems range from 11" to 14". Resolution ranges from 640 x 480 Pixels (horizontal measurement first and then vertical) to 1600 x 1200 Pixels. Each Pixel in the monitor is made of a trio dots, one each for the colours red, blue and green.

An average monitor is capable of refreshing 60 times per second (60Hz), whereas higher quality monitors might refresh at 100Hz. The refresh rate measures how often the display of the screen is redrawn from the contents of the video adapter memory.

Both the resolution and refresh rate of the monitor are tied into the capability of the system video adapter. Most monitors are capable of supporting several different resolutions and refresh rates (with the common exception of LCD screens in portables).

CHAPTER THREE

3.1 BASIC TROUBLESHOOTING GUIDELINES

Before starting any system troubleshooting, there are a few basic steps that should be performed to ensure consistent starting point and to allow isolating the failed component.

1. Turn off the system and any peripherals from devices. Disconnect all external peripherals from the system except for the keyboard and the video display.
2. Make sure the system is plugged in to a properly grounded power outlet.
3. Make sure the keyboard and the video displays are connected to the system. Turn on the video display, and turn up the brightness and contrast controls to at least two thirds of the maximum. Some displays have onscreen controls that may not be intuitive. Consult the display documentation for more information on how to adjust these settings.
4. To allow the system to boot from a hard disk, make sure there is no floppy disk in the floppy with DOS or diagnostics on it in the floppy drive for testing.
5. Turn on the system. Observe the power supply and chassis fan (if any), and the lights on either the system front panel or power supply. If the fans don't spin and the lights don't light, the power supply or the motherboard might be defective.
6. Observe the power-on-self-test (POST). If no errors are detected, the system beeps once and boot up. Errors that display on the screen (non-fatal errors) and which do not lock up the system display a text message that varies according to BIOS type and version. Record any error that occur and refer to the codes for information. Errors that lock up the system (fatal error) are indicated by a series of audible beeps.
7. Confirm that the operating system loads successfully.

Problems during the POST

Problems that occur during the POST are usually caused by incorrect hardware configuration or installation. Actual hardware failure is a far less frequent cause.
If you have a POST error, check the following:

1. Are all cables correctly connected and secured?
2. Are the configuration settings correct in set up?
3. Are all drivers properly installed?
4. Are switches and jumpers on the baseboard correct, if changed from the default setting?

3.2 THE TROUBLESHOOTING PROCESS

Regardless of how complex a particular computer or peripheral device may be, a dependable troubleshooting procedure can be broken down into 4 basic steps:

(1) Define the symptoms.
(2) Identify and isolate the potential source (or Location) of the problem.
(3) Repair or replace the suspected component or assembly.
(4) Retest the unit thoroughly to be sure that the problem has been solved. If the problem is not solved start again from the computer troubleshooting, not just for pc memory systems.

DEFINE THE SYMPTOM:

A firm understanding of all symptoms is needed before opening up the system. The following should be consider:

Is the disk or tape inserted properly? Is the power or access LED fit? Does the problem occur only when the computer is tapped or moved? By recognizing and understanding the understanding the symptoms, it can be much easier to trace a problem to the appropriate assembly or component.

The symptom should be written down. The note taking may seen tedious, but a written record of symptom and circumstance will help keep fit on the task at hand.

IDENTIFY AND ISOLATE:-

Before trying to isolate a problem within a piece of computer hardware, it's good to be first sure that it is the test equipment itself that is causing the problem. It should be remember that storage device work because of an intimate mingling of hardware and software. A faulty or improperly configured piece of software can cause configuring system errors.

After configuring that the failure lies in the system's hardware, the possible problem can be identified .The actual repair process can only begin after the potential problem area is identified and it possibly tracking of the fault to components level.

REPAIR OR REPLACE

After understanding what is wrong and where to look, the actual repair process can now begin which will correct the symptoms. Most storage devices are a mix of both electronic circuit and electromechanical devices, so most procedures will require the exchange of electronic or electromechanical parts. As a general rule, all procedures should be considered important and should be careful.

Parts are usually classified as components subassemblies. Components can serve many different purposes in a computer Resistors, capacitors, transformers, monitor, and ICs are just a few type of component part, monitor. and ICs are just a few type of component parts within themselves a detective component must be replaced. A subassembly can be composed of many individual components. Unlike components sub assembling serve a single specific purpose in a storage device (i.e. in read / write head or amplifier PC board), but it can usually be repaired by locating and replacing any faulty component. For example, it's always advisable to replace, a faulty keyboard than repair it, also monitor are always being advised to replace than repair.

RETEST

When a repair is finally complete, the system must be reassembled carefully before testing it. Guards, housings, cables and shields must be restored before final testing. If the symptoms persist, the symptoms need to be re-evaluated and narrow the problem to another part of the equipment. If normal operation is restored (or greatly improved the computer's items functions should be tested.

Storage devices are just a collection of assemblies and each assembly is a collection of part. Normally, everything works together, but when one part fails it may cause one or more interconnected part to fail as well.

Several repair attempts should be made while repairing a computer, be prepared to make several attempts before the computer is repaired completely.

POST REPAIR TESTING

One should be familiar with the idea of "burn-in" testing once the repair is complete. In modern electronics, the general rule of reliability suggests that a device will fail very quickly or will last a long time. With this rule in mind, it is usually desirable to run the PC for at least 24 hours after the repair is done. This will "stress" the new part or drive, and (hopefully) any marginal part will fail right there while the system is still on the workbench. This kind of testing improves the reliability of the repair.

3.3 DIAGNOSTIC SOFTWARE

Typical software tool include the commercial standard like DEFRAG, SCANDISK and CHKDSK.

- <u>DATA_REG.ZIP</u>

A hard disk crash is frustrating and frightening. Even with the work fully backed up the time and expense of troubleshooting and disk replacement can be an expense proposition. Recovering data from a crashed hard disk is more of an art than a science, and there are few tools that are actually up to the task, but Tiramisu is one of the only hardware tools available that can reconstruct data from failed drives. Tiramisu is designed to help devices that have been hit by a virus, scratched by a head crash, accidentally formatted or partitioned, corrupted by a power failure, or damaged by buggy applications. Tiramisu scans the disk even when there are physical damages – the found data are analyzed and reconstructed. The program works on drives without a readable boot sector, readable FAT, or readable directory entries.

➢ DUGIDE.ZIP

Every hard drive ever manufactured has its own "geometry" – the number of cylinders, heads, sectors and so on, which defines the way a drive retain data. When a new drive is installed, or old drive is replaced, or the contents of CMOS are lost, and one is faced with the prospect of tracking down and restoring the geometry figures for each hard drive. Programs like DUG_IDE.EXE integrate a hard drive and display the resulting details for quick reference.

➢ AUTO TEST. ZIP

Part of device testing often involves measuring performance, while performance is not a critical factor with floppy drives, poor performance (usually in conjunction with other DOS errors) may suggest a failing device or the presence of a hardware conflict. The AUTOTEST.EXE diagnostic is designed to measure floppy drive performance by putting the drive through a series of random and sequential reads. Unusually long read time may suggest problems with the hardware heads. track stepping motor, or spindle motor.

➢ DFR.ZIP

There are some occasions where the data on a floppy disk is just as valuable as the data on a hard drive. When the floppy diskette fails the loss of program on it may be catastrophic. The DFR package mark Vitt is a highly automated recovery tool that specializes in recovering files from floppy disks. DFR can recover files to hardware or another floppy disk.

➢ CD SPEED. ZIP

The performance of a CD-ROM drive may eventually need to be measured and evaluated during the course of troubleshooting or upgrade procedure. The CD SPEED. ZIP package provides a tool that measure the sustained data transfer rate of a CD ROM drive. the efficiency of the CD ROM device in services, and the overall utilization of CPU processing power (which helps determines just how much of a load the CD-ROM device actually is on the system).

➢ CDTA.ZIP

Timing and performance are vital aspects of all drives. Thus there is need to measure the performance and data throughput of new or replacement CD-ROM device. The CDTA.ZIP package is designed to test the data throughputs and access times for most CD ROM drives using conventional low –level devices and MSCDEX.

➢ CHECK136.ZIP

CHECKSYS is a basic system diagnostic ıat abandons the use of menu screens and windows in favour of a ommand-line system. When CHECKSYS is executed, it will return adescription of the whole PC system (or an error level if necessary). CHKSYS can also be used to inspect specific systems and return one line description of a system status. The command-line architecture of CHECKSYS also makes it possible to add routine diagnostic functions in the system start up files (i.e. AUTO EXEC. BAT), or other batch files.

- CONF810E .ZIP

Often lies in knowing "what's in the box" Rather than the time consuming process of disassembling the computer and examining each item by eye, a well-written up to the minute system information utility can tell everything one needs to know before even picking up a screwdriver. The long-running PC- config series of utilities by Michael Holin is one of the most recognized and respected shareware system information utilities benchmarking programs available.

3.4 GENERAL REQUIREMENT FOR SYSTEM SOFTWARE

As end-user requirements becomes more exacting, collection of programs grow in complexity and size. The following requirements were taken into consideration in the developing of the program.

1. Correctness and reliability: These outweigh all the other requirements. Fault in the program were taken into consideration and corrected so that logic errors can be avoided, as the computer cannot detect it.
2. Flexibility and reversibility: The program was design in such a way that it can easily be modified.
3. Efficiency: As a software system grows in size and complexity, it also tends to run slower. Slow response time may be quite time consuming and strenuous but efficiency was not pursued at the expense of correctness and only in extreme

cases at the expense of flexibility. The program is quite efficient as it minimize time and maximizes efficiency.

3.5 SYSTEM ANALYSIS

The object of the analysis phase is to find out exactly what the end-user wants the final program to do (of course the programmer himself may be the end-user). The problem is often that users do not know what to expect from the program. The program has been written in such away that the end-user knows what the program is expected to perform.

3.6 DESIGN OBJECTIVE

The users were taken into consideration by making the system user-friendly, through the use of interactive mode. The use of custom screens for economic reason was also adopted.

Objectives of the designing of the system are:

a) To list all the necessary files that would be used.
b) To keep logbook of all successful operation on computers.
c) To save status of user and nature of work done on system in the program using the consultation form.
d) To develop procedures and programs.
e) To identify how all programs will work together as a system.

3.7 DESIGN SPECIFICATION

After gathering and analysing data, an explicit specification is written which set forth for what the system should be doing. Almost, always, the system specifications begin with a detailed description of the required output of the system.

However, once the system's output is specified, it then becomes necessary and important to determine the input data, which is known as input requirements. These requirements specify the source of the data needed by the system, as well as the formats in which the data are most easily and economically retrieved.

3.8 INPUT DESIGN

Basic Inputs to be used in the system are as follows:

1. TCONSULT

S/NO	FIELD NAME	TYPE	WIDTH	FUNCTION
1.	Cust_ name	Character	30	To hold the name of the customer.
2.	Contact_ add	Memo	4	To hold the contact address of the customer.
3.	Contact_ phone	Numeric	10	To hold the contact phone number of the customer.
4.	Office_add	Memo	4	To hold the office address of the customer.
5.	Office_ phone	Numeric	10	To hold the office phone number of the customer.
6.	Prob_descript	Character	10	To hold the description of the problem.

7.	Amt _ char	Numeric	20.2	To hold the amount charged on the problem.
8.	Mode_ pay	Character	10	To describe the mode of payment.
9..	Mach _ No	Numeric	20	To hold the serial number of the machines

2. Ttro

S/N	FIELD NAME	TYPE	WIDTH	FUNCTION
1.	User _name	Character	30	To hold the name of the user
2.	Date in	Date	8	To hold the current date which the work was brought.
3.	Date out	Date	8	To hold the date on which the work is to be completed and returned.
4.	Machine No	Character	20	To hold the serial number of machine
5.	Prob _ descript	Memo	4	To hold the description of the problem.
6.	Solution	Memo	4	To hold the solution to the problem.
7.	Rem	Memo	4	To hold the remark about the work.
8.	User add	Memo	4	To hold the user's address
9.	User phone	Numeric	10	To hold the phone number of the user.

3.9 OUTPUT DESIGN

Output refers to the result and information that are generated by a system. The output from a computer system is required primarily to communicate the result of processing to users or other systems or more importantly, to provide a permanent (hard) copy of this result for consultation and for the technician/ engineers requirement.

However, the output specification is a determinant for the input specification. The design process of the output design begins by the identification of the output the system must produce. It is as a result of this that in designing output for the maintenance and troubleshooting software, that the needs of the end-users were fully considered.

Specifically, the following reports were expected to be generated from the output specification. They include:

1. The different components of a PC system.
2. The description of problem as reported by the client.
3. The fault as diagnosed by the engineer
4. Report on general maintenance and troubleshooting guidelines.

3.10 HARDWARE REQUIREMENT

Hardware is the physical components in a computer system, i.e. computer peripherals. The following are the list of recommended specifications, which if adhered to would produce the best result, and the system would be free from error/fault:

a) Monitor (display)

b) Keyboard
c) Mouse
d) UPS (interruptible power supply)
e) System unit which comprise of:
   i. System case (chassis)
   ii. Motherboard (system board)
   iii. Microprocessor
   iv. Memory (RAM)
   v. Power supply
   vi. Floppy drive
   vii. Hard disk
   viii. CD-ROM drive.

3.11 SOFTWARE REQUIREMENT

Software requirements are the basic and other relevant application software that is needed to be installed on the computer system in order to make maximum utilization of the computer system and the developed software. It is because of this fact that the following software must be installed on the system.

The software include:

A. Operating system software (Windows 98 or 2000)
B. Designed software
C. Other application software such as:
   Ms Word
   Ms Excel
   Ms PowerPoint
   Visual Studio

CHAPTER FOUR

4.1 THE PROCESS OF SOFTWARE DEVELOPMENT

The process of software development is a cooperative effort of the users of software and computer professionals. While computer professionals are those familiar with the technology and how it can be applied to meet a business information processing needs, users of software (Non-Technicians) on this side have an in-depth familiarity with their respective functional areas.

The skills and knowledge of these two groups complement each other and can be combined to create any type of information system during the software development process.

However, because system or software development is a team effort, most organizations have adopted a standardized "system/software Development Methodology" that provides a framework for cooperation and a successful development of a new system. This step-by-step development procedure is illustrated below in a tabular form.

STEPS	ACTIVITIES	MEASURABLE OUTPUT
Analysis & Specification	Appraisal of existing situation Identification of users requirement	
Design	Design the overall program structure. Design the detailed processor processing logic	Process logic specifications, using pseudo code or flowchart techniques
Programming	Writing of code in the appropriate processor language Entry of written of code into the computer	Hand written program code Compute printed list out code

Testing and Debugging	Removal of syntax and logic errors. Final testing of program.	Error free program execution.
Installation and Maintenance	Error free program installation User training	Program documentation and upgrade procedures.

However, software development process is essentially the same, be it for inventory management system, airline reservation system etc. As a member of a “Project Team” progress through the procedures outlined in a software development methodology, the result of one step provide the input for the next step and/or subsequent steps.

Hence for a successful program development of ***MAINTENANCE/TROUBLESHOOTING*** software, it is assumed that the writer of the project work is part of the project team as a programmer. The above five (5) steps in the table were followed one after the other in order to design and develop a customized software package called the *“**MAINTENANCE/TROUBLESHOOTING** Software”*.

4.2 CHOICE OF PROGRAMMING LANGUAGE

From the previous analysis it is pertinent to say that the proposed system is going to be used to solve various malfunctioning problems and to take note of various jobs brought into the company and the probable solution in each case. Due to this fact, the choice of the programming language chosen for the development of the system is Database Management System (DBMS) package with special preference for Visual FoxPro.

4.3 FEATURES OF CHOSEN PROGRAMMING LANGUAGE

However, the choice of Visual FoxPro is because of the following reasons and features possess by the application software:

- It is easy to write an active user interface program and also simple to understand.
- It is user – friendly
- It reduces data redundancy
- Data integrity can be maintained
- Provide easy and greater access to information
- Individual database file can be designed to meet specific requirement of particular functional unit of an organization
- Visual FoxPro allows the source program to be compiled to an executable file thereby allowing the program to be run independently of the application software that is used in coding the program.

4.4 PROGRAM TESTING AND IMPLEMENTATION.

The design of a system is never complete until it has been tested, implemented and finally documented. The implementation phase is basically meant for the user of the system and it entails translating the ideas and details plans developed in the system study into actual practices. It is also the occasion for effecting the end view for the system, to see it functioning and to monitor its procedures.

Efforts have been made to ensure that the program is working as desired and that the system will function as proposed. The system package has been developed in an easy and interactive way to make the implementation of the system comprehensive. All the necessary information within each module has been taken into consideration, thus providing a menu driven for the operation to be performed on each menu option.

4.5 IMPLEMENTATION TOOLS.

The system is an integrated user friendly and menu driven package. It has been developed as a maintenance trouble-shooting guide for any user.

The application package is written in visual fox pro programming language.

4.6 SYSTEM DOCUMENTATION.

Documentation is virtually important if a program is to be understood, not only by others but also by the original programmer at a later stage. So any program should be documented while being written. That is, the program itself should contain comments or remarks stating exactly what various parts of the program are to do. This technical documentation is of utmost importance to the programmer (or other programmers who are working with the program) when the time comes to debug, modify, or update the program.

As a result of this, the program was well documented. The following are also part of the system documentation.

1. Program flowchart
2. Hardware specification
3. Program listing.

4.7 USER'S MANUAL.

The software system is user-friendly. Software is said to be user-friendly if its human users find it easy to use. The use of user-interface such as windows, an important component of user-friendliness enhances this.

However, the user still needs a written manual that guides him/her through the software system so that appropriate input should be supplied to the software system for the system to work effectively.

Below is the guideline to the software system.

4.7.1 USING THE SYSTEM.

The software system could be run on a floppy disk or on the hard disk by installing the software on the hard disk.

4.8 PROGRAM INSTALLATION

This is the process of transferring the developed system from the floppy disk to permanent storage device called the Hard Disk.

However, due to vast improvement in recent operating systems, the installation of the software is a very simple operation. The procedure as follows:

Steps	Procedure	Result
Go to start	Click	Start pop menu is displayed
Programs submenu	Click	Programs submenu is displayed
Windows explorer	Click	Windows explorer window
Insert the diskette containing The Maintenance/Troubleshooting Software into A.		
Create a folder	Click file, then new folder	
Highlight the content of the diskette	Click the first icon, then hold down the shift key	the whole document of A is highlighted, and click the last icon

To copy	right click inside The highlighted icon For some pop up menu And select the command
Destination	copy the content first to C:/

At the end of the installation, the floppy diskette becomes a backup and should be well safeguarded against any damage for future use.
To start the system after successful installation of the software, all one need to do is to go to start and run the folder name that have been created.

4.9 PROGRAM LAYOUT DESIGN

The main menu of the maintenance / Troubleshooting is in form of rectangular square containing pop up pulls down menu options. The user of the system is expected to use the mouse or up and down arrow key to highlight the option. This is preceded by either enter key or double clicking to carry out the execution. The main program menu is as shown in the appendix B (fig. 1-4).

CHAPTER FIVE

5.1 SUMMARY

The aims and objective of this project is to far fetch the implication of computer application to Maintenance/Troubleshooting tasks carried out in AMI-TECH computers (Nig.) Limited.

The maintenance/troubleshooting tasks which is often carried out on a regular basis in the company either for the Client or the company is lacking in some aspect as a result of poor documentation of specific errors and possible solutions.

It was in view of this that customized software was suggested and undertaken to enable the users carry out their routine maintenance/troubleshooting tasks. This method of computerization will completely reduce if not eliminate the burden of guess work or suspending the system until an engineer comes when a particular error occurs.

5.2 CONCLUSION

Without any hesitations, the computer technology has created an overwhelming awareness for itself.

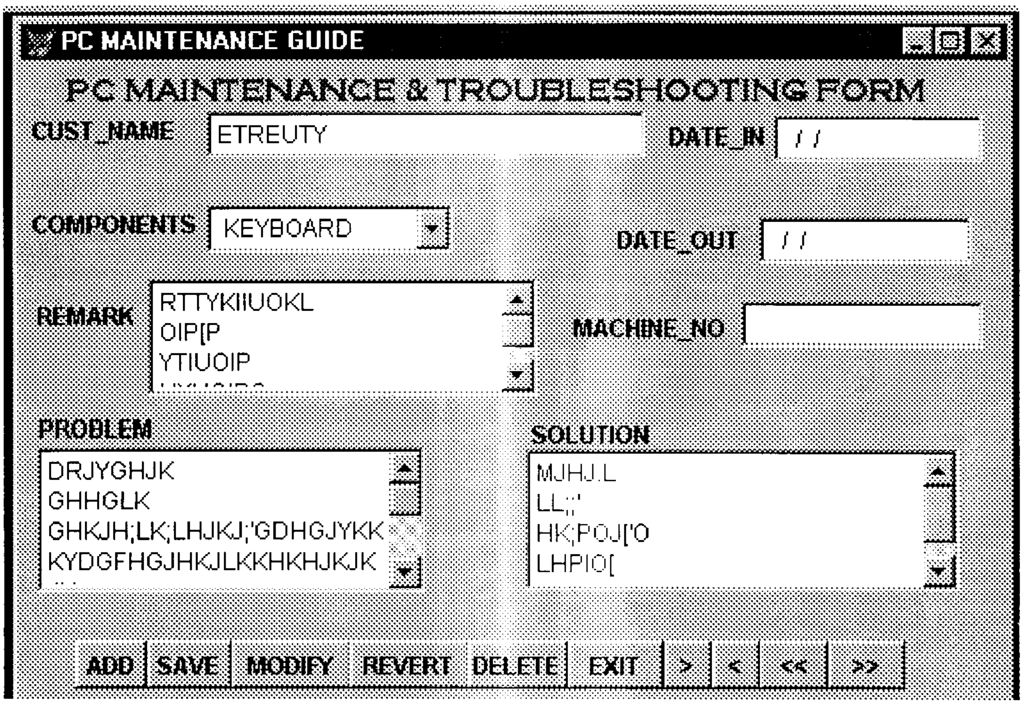
When one looks back, with an analytical mind, one can conclude that computer technology keeps on advancing with remarkable increase in speed, accuracy and reliability. One can now access a computer system, take a look at the error, note down the error and proceed to correct the error without necessarily looking for a technician.

It is because of these important features of the computer system that the project was undertaken to lighten the burden of packing up a system when fault developed.

5.3 RECOMMENDATION

The working condition of any computer system determines its efficiency and reliability. It is based on this fact that the followings were recommended:

- The consultants in the maintenance workshop of AMI-TECH computers (Nig.) Ltd., should make sure a particular fault /complaint about a system is properly noted i.e. documented.
- The maintenance /troubleshooting software be implemented in the company in such a way that any of the staff can attend to any task at any time irrespective of his/her background.



PC MAINTENANCE & TROUBLESHOOTING FORM

CUST_NAME ETREUTY

DATE_IN / /

COMPONENTS KEYBOARD

DATE_OUT / /

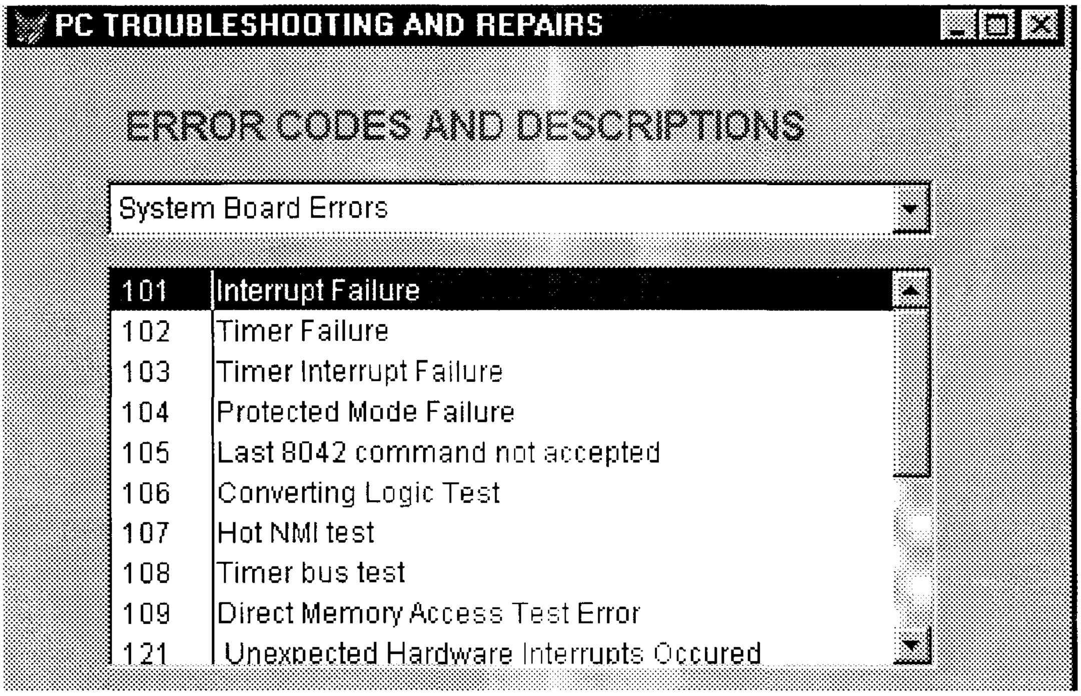
REMARK
RTTYKIIUOKL
OIP[P
YTIUOIP

MACHINE_NO

PROBLEM
DRJYGHJK
GHHGLK
GHKJH;LK;LHJKJ;'GDHGJYKK
KYDGFHGJHKJLKKHKHJKJK

SOLUTION
MJHJ.L
LL;;'
HK;POJ['O
LHPIO[

FIGURE 3



System Board Errors

101	Interrupt Failure
102	Timer Failure
103	Timer Interrupt Failure
104	Protected Mode Failure
105	Last 8042 command not accepted
106	Converting Logic Test
107	Hot NMI test
108	Timer bus test
109	Direct Memory Access Test Error
121	Unexpected Hardware Interrupts Occured

FIGURE 4

APPENDIX A

```
PUBLIC oform1

oform1=NEWOBJECT("form1")
oform1.Show
RETURN


        **************************************************
*-- Form:        form1 (c:\nknew\ffirst.scx)
*-- ParentClass:  form
*-- BaseClass:    form
*
DEFINE CLASS form1 AS form


        DataSession = 1
        Top = 15
        Left = 12
        Height = 314
        Width = 535
        Desktop = .T.
        DoCreate = .T.
        Picture = "ph02398u.bmp"
        Caption = "PGD COMPUTERS"
        FontBold = .T.
        FontSize = 12
        MaxButton = .F.
        MinButton = .F.
        Icon = "..\program files\microsoft visual studio\vfp98\"
        WindowType = 1
        WindowState = 0
        ForeColor = RGB(0,0,255)
        BackColor = RGB(0,0,0)
        FillColor = RGB(255,0,255)
        Name = "Form1"


        ADD OBJECT label1 AS label WITH ;
                AutoSize = .T., ;
                FontBold = .T., ;
                FontName = "Modern No. 20", ;
                FontSize = 18, ;
                BackStyle = 0, ;
                Caption = "SYSTEMATIC APPROACH TO ", ;
                Height = 28, ;
                Left = 96, ;
                Top = 24, ;
                Width = 356, ;
                ColorSource = 4, ;
                ForeColor = RGB(0,255,64), ;
                Name = "Label1"


        ADD OBJECT label3 AS label WITH ;
                AutoSize = .T., ;
                FontBold = .T., ;
                FontName = "Modern No. 20", ;
                FontSize = 11, ;
                BackStyle = 0, ;
                Caption = "(A CASE STUDY OF AMI-TECH COMPUTERS MAINTENANCE
DEP'T)", ;
                Height = 20, ;
                Left = 8, ;
                Top = 94, ;
                Width = 521, ;
                ForeColor = RGB(0,255,0), ;
                Name = "Label3"


        ADD OBJECT label4 AS label WITH ;
                AutoSize = .T., ;
                FontBold = .T., ;
                FontItalic = .F., ;
                FontSize = 14, ;
                BackStyle = 0, ;
                Caption = "A project developed by:", ;
                Height = 25, ;
                Left = 12, ;
                Top = 132, ;
                Width = 222, ;
                ForeColor = RGB(255,255,255), ;
                BackColor = RGB(128,0,64), ;
                Name = "Label4"


        ADD OBJECT label5 AS label WITH ;
                AutoSize = .T., ;
                FontBold = .T., ;
                FontItalic = .F., ;
                FontSize = 14, ;
                BackStyle = 0, ;
                Caption = "IDEM NKOYO", ;
                Height = 25, ;
                Left = 240, ;
                Top = 156, ;
                Width = 126, ;
                ForeColor = RGB(255,255,255), ;
                BackColor = RGB(0,64,128), ;
                Name = "Label5"


        ADD OBJECT label6 AS label WITH ;
                AutoSize = .T., ;
                FontBold = .T., ;
                FontItalic = .F., ;
                FontSize = 12, ;
                BackStyle = 0, ;
                Caption = "PGD/MCS/98/99/ 807      ", ;
                Height = 22, ;
                Left = 228, ;
                Top = 180, ;
                Width = 180, ;
        ForeColor = RGB(255,255,255), ;
        BackColor = RGB(0,64,128), ;
        Name = "Label6"


ADD OBJECT label7 AS label WITH ;
        AutoSize = .T., ;
        FontBold = .T., ;
        FontItalic = .F., ;
        FontSize = 11, ;
        BackStyle = 0, ;
        Caption = "MATHS/COMPUTERS DEPT", ;
        Height = 20, ;
        Left = 204, ;
        Top = 204, ;
        Width = 202, ;
        ForeColor = RGB(255,255,255), ;
        BackColor = RGB(0,64,128), ;
        Name = "Label7"


ADD OBJECT timer1 AS timer WITH ;
        Top = 144, ;
        Left = 168, ;
        Height = 23, ;
        Width = 23, ;
        Interval = 5, ;
        Name = "Timer1"


ADD OBJECT timer2 AS timer WITH ;
        Top = 144, ;
        Left = 240, ;
        Height = 23, ;
        Width = 23, ;
        Interval = 3200, ;
        Name = "Timer2"


ADD OBJECT label8 AS label WITH ;
        AutoSize = .T., ;
        FontBold = .T., ;
        FontItalic = .F., ;
        FontSize = 10, ;
        BackStyle = 0, ;
        Caption = "FEDERAL UNIVERSITY OF TECHNOLOGY", ;
        Height = 18, ;
        Left = 192, ;
        Top = 228, ;
        Width = 257, ;
        ForeColor = RGB(255,255,255), ;
        BackColor = RGB(0,64,128), ;
        Name = "Label8"


ADD OBJECT label9 AS label WITH ;
```

```
        AutoSize = .T., ;
        FontBold = .T., ;
        FontItalic = .F., ;
        FontSize = 11, ;
        BackStyle = 0, ;
        Caption = "BOSSO, MINNA", ;
        Height = 20, ;
        Left = 252, ;
        Top = 252, ;
        Width = 113, ;
        ForeColor = RGB(255,255,255), ;
        BackColor = RGB(0,64,128), ;
        Name = "Label9"


ADD OBJECT label2 AS label WITH ;
        FontBold = .T., ;
        FontName = "Copperplate Gothic Bold", ;
        FontSize = 14, ;
        BackStyle = 0, ;
        Caption = "PC MAINTENANCE AND TROUBLESHOOTING", ;
        Height = 36, ;
        Left = 28, ;
        Top = 60, ;
        Width = 490, ;
        ForeColor = RGB(0,255,0), ;
        Name = "Label2"


ADD OBJECT timer3 AS timer WITH ;
        Top = 132, ;
        Left = 300, ;
        Height = 23, ;
        Width = 23, ;
        Interval = 5000, ;
        Name = "Timer3"


PROCEDURE Click
        *release thisform
ENDPROC


PROCEDURE Load

        public maxw,maxh,wid,hei

        store 0 to maxw,maxh,wid,hei

        maxw = _screen.left + _screen.width
        maxh = _screen.top + _screen.height

        wid = thisform.left
        hei = thisform.top
ENDPROC
```

```
        PROCEDURE timer1.Timer

                if (wid + thisform.left <= maxw)
                thisform.left = thisform.left + 1
                else
                thisform.left = -15
                endif
        ENDPROC


        PROCEDURE timer2.Timer
                thisform.release()
        ENDPROC


        PROCEDURE timer3.Timer
                THISFORM.RELEASE()
        ENDPROC


ENDDEFINE
*
*-- EndDefine: form1
**************************************************



PUBLIC oform1

oform1=NEWOBJECT("form1")
oform1.Show
RETURN


        **************************************************
*-- Form:        form1 (c:\nknew\fdescipt1.scx)
*-- ParentClass:  form
*-- BaseClass:    form
*
DEFINE CLASS form1 AS form


        Top = 30
        Left = 96
        DoCreate = .T.
        Caption = "PC MAINTENANCE & TROUBLESHOOTING "
        FontBold = .T.
        Name = "Form1"


        ADD OBJECT cbotopic AS combobox WITH ;
                Comment = "", ;
                RowSourceType = 6, ;
                RowSource = "tchap1.topic", ;
                ControlSource = "tchap1.topic", ;
                Height = 24, ;
                Left = 103, ;
                TabIndex = 2, ;
                Top = 7, ;
                Width = 187, ;
                Name = "cboTopic"


        ADD OBJECT lbltopic AS label WITH ;
                AutoSize = .T., ;
                FontBold = .T., ;
                WordWrap = .T., ;
                BackStyle = 0, ;
                Caption = "COMPONENTS", ;
                Left = 13, ;
                Top = 13, ;
                Width = 82, ;
                TabIndex = 1, ;
                Name = "lblTopic"


        ADD OBJECT edtdesc AS editbox WITH ;
                Comment = "", ;
                Height = 180, ;
                Left = 48, ;
                TabIndex = 6, ;
                Top = 43, ;
                Width = 312, ;
                ControlSource = "tchap1.desc", ;
                Name = "edtDesc"


        PROCEDURE cbotopic.InteractiveChange
                thisform.refresh()
        ENDPROC


ENDDEFINE
*
*-- EndDefine: form1

PUBLIC oform1

oform1=NEWOBJECT("form1")
oform1.Show
RETURN


        **************************************************
*-- Form:        form1 (c:\nknew\fconsult2.scx)
*-- ParentClass:  form
*-- BaseClass:    form
*
DEFINE CLASS form1 AS form
```

```
Top = 16
Left = -74
Height = 259
Width = 457
DoCreate = .T.
Caption = "AMI-TECH CONSULTAION FORM"
Name = "Form1"


ADD OBJECT txtcust_name AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tconsult.cust_name", ;
        Format = "!", ;
        Height = 23, ;
        Left = 92, ;
        MaxLength = 30, ;
        TabIndex = 2, ;
        Top = 49, ;
        Width = 291, ;
        Name = "txtCust_name"


ADD OBJECT lblcust_name AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "CUST_NAME", ;
        Left = 5, ;
        Top = 55, ;
        Width = 73, ;
        TabIndex = 1, ;
        Name = "lblCust_name"


ADD OBJECT edtcust_add AS editbox WITH ;
        Comment = "", ;
        Height = 54, ;
        Left = 216, ;
        TabIndex = 4, ;
        Top = 90, ;
        Width = 168, ;
        ControlSource = "tconsult.cust_add", ;
        Name = "edtCust_add"


ADD OBJECT lblcust_add AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "CUST_ADDRESS", ;
        Left = 216, ;
        Top = 78, ;
        Width = 98, ;
        TabIndex = 3, ;
        Name = "lblCust_add"
```

```
ADD OBJECT txtcust_phone AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tconsult.cust_phone", ;
        Format = "999-999999", ;
        Height = 23, ;
        Left = 350, ;
        MaxLength = 10, ;
        TabIndex = 6, ;
        Top = 146, ;
        Width = 96, ;
        Name = "txtCust_phone"


ADD OBJECT lblcust_phone AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "CUST_PHONE", ;
        Left = 264, ;
        Top = 156, ;
        Width = 83, ;
        TabIndex = 5, ;
        Name = "lblCust_phone"


ADD OBJECT edtprob_descript AS editbox WITH ;
        Comment = "", ;
        Format = "!", ;
        Height = 53, ;
        Left = 5, ;
        TabIndex = 8, ;
        Top = 91, ;
        Width = 163, ;
        ControlSource = "tconsult.prob_descript", ;
        Name = "edtProb_descript"


ADD OBJECT lblprob_descript AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "PROB_DESCRIPTION", ;
        Left = 5, ;
        Top = 76, ;
        Width = 123, ;
        TabIndex = 7, ;
        Name = "lblProb_descript"


ADD OBJECT txtamt_char AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tconsult.amt_char", ;
        Format = "9,999,999,999,999.00", ;
        Height = 23, ;
        Left = 92, ;
        TabIndex = 10, ;
        Top = 149, ;
        Width = 149, ;
        Name = "txtAmt_char"


ADD OBJECT lblamt_char AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "AMT CHARGED", ;
        Left = 5, ;
        Top = 160, ;
        Width = 86, ;
        TabIndex = 9, ;
        Name = "lblAmt_char"


ADD OBJECT lblp_mode AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "PAY_MODE", ;
        Left = 242, ;
        Top = 187, ;
        Width = 64, ;
        TabIndex = 11, ;
        Name = "lblP_mode"


ADD OBJECT txtmach_no AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tconsult.mach_no", ;
        Height = 23, ;
        Left = 92, ;
        MaxLength = 10, ;
        TabIndex = 14, ;
        Top = 188, ;
        Width = 81, ;
        Name = "txtMach_no"


ADD OBJECT lblmach_no AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "MACH_N0", ;
        Left = 5, ;
        Top = 189, ;
        Width = 57, ;
        TabIndex = 13, ;
        Name = "lblMach_no"


ADD OBJECT txtdate AS textbox WITH ;
        Comment = "", ;
```

```
		ControlSource = "tconsult.date", ;
		Height = 23, ;
		Left = 49, ;
		TabIndex = 16, ;
		Top = 23, ;
		Width = 73, ;
		Name = "txtDate"


	ADD OBJECT lbldate AS label WITH ;
		AutoSize = .T., ;
		WordWrap = .T., ;
		BackStyle = 0, ;
		Caption = "DATE", ;
		Left = 5, ;
		Top = 27, ;
		Width = 31, ;
		TabIndex = 15, ;
		Name = "lblDate"


	ADD OBJECT label1 AS label WITH ;
		FontBold = .T., ;
		FontName = "Comic Sans MS", ;
		FontSize = 14, ;
		BackStyle = 0, ;
		Caption = "CONSULTATION FORM", ;
		Height = 36, ;
		Left = 142, ;
		Top = 0, ;
		Width = 242, ;
		ForeColor = RGB(0,0,255), ;
		Name = "Label1"


	ADD OBJECT cmdgrpeditor AS commandgroup WITH ;
		AutoSize = .T., ;
		ButtonCount = 6, ;
		BackStyle = 1, ;
		Value = 1, ;
		Height = 34, ;
		Left = 96, ;
		Top = 220, ;
		Width = 122, ;
		TabIndex = 7, ;
		BackColor = RGB(192,192,192), ;
		Name = "cmdgrpeditor", ;
		Command1.AutoSize = .F., ;
		Command1.Top = 5, ;
		Command1.Left = 5, ;
		Command1.Height = 23, ;
		Command1.Width = 28, ;
		Command1.Caption = "\<Add", ;
		Command1.Name = "cmdadd", ;
		Command2.AutoSize = .F., ;
		Command2.Top = 6, ;
```

```
        Command2.Left = 34, ;
        Command2.Height = 23, ;
        Command2.Width = 28, ;
        Command2.Caption = "\<Save", ;
        Command2.ColorScheme = 2, ;
        Command2.Name = "cmdsave", ;
        Command3.AutoSize = .F., ;
        Command3.Top = 5, ;
        Command3.Left = 61, ;
        Command3.Height = 23, ;
        Command3.Width = 28, ;
        Command3.FontBold = .T., ;
        Command3.Caption = "\<Del", ;
        Command3.Name = "cmddelete", ;
        Command4.AutoSize = .F., ;
        Command4.Top = 5, ;
        Command4.Left = 89, ;
        Command4.Height = 23, ;
        Command4.Width = 28, ;
        Command4.FontBold = .T., ;
        Command4.Caption = "E\<xit", ;
        Command4.Name = "cmdexit", ;
        Command5.AutoSize = .F., ;
        Command5.Top = 5, ;
        Command5.Left = 5, ;
        Command5.Height = 23, ;
        Command5.Width = 28, ;
        Command5.FontBold = .T., ;
        Command5.Caption = "\<Rev", ;
        Command5.Name = "cmdrevert", ;
        Command6.AutoSize = .F., ;
        Command6.Top = 5, ;
        Command6.Left = 34, ;
        Command6.Height = 23, ;
        Command6.Width = 28, ;
        Command6.FontBold = .T., ;
        Command6.Caption = "\<Mod", ;
        Command6.Name = "cmdmodify"


ADD OBJECT cmdgrpnavigator AS commandgroup WITH ;
        AutoSize = .T., ;
        ButtonCount = 4, ;
        BackStyle = 1, ;
        Value = 1, ;
        Height = 33, ;
        Left = 216, ;
        Top = 220, ;
        Width = 120, ;
        TabIndex = 8, ;
        BackColor = RGB(192,192,192), ;
        Name = "cmdgrpnavigator", ;
        Command1.AutoSize = .F., ;
        Command1.Top = 5, ;
        Command1.Left = 5, ;
        Command1.Height = 23, ;
        Command1.Width = 27, ;
        Command1.FontBold = .T., ;
        Command1.Caption = "<<", ;
        Command1.Name = "cmdtop", ;
        Command2.AutoSize = .F., ;
        Command2.Top = 5, ;
        Command2.Left = 32, ;
        Command2.Height = 23, ;
        Command2.Width = 27, ;
        Command2.FontBold = .T., ;
        Command2.Caption = "<", ;
        Command2.Name = "cmdprevious", ;
        Command3.AutoSize = .F., ;
        Command3.Top = 5, ;
        Command3.Left = 59, ;
        Command3.Height = 23, ;
        Command3.Width = 27, ;
        Command3.FontBold = .T., ;
        Command3.Caption = ">", ;
        Command3.Name = "cmdnext", ;
        Command4.AutoSize = .F., ;
        Command4.Top = 5, ;
        Command4.Left = 88, ;
        Command4.Height = 23, ;
        Command4.Width = 27, ;
        Command4.FontBold = .T., ;
        Command4.Caption = ">>", ;
        Command4.Name = "cmdbottom"


    ADD OBJECT cbop_mode2 AS combobox WITH ;
        Comment = "", ;
        RowSourceType = 1, ;
        RowSource = "INSTALLMENTALLY, FULL", ;
        ControlSource = "tconsult.p_mode", ;
        Height = 24, ;
        Left = 312, ;
        TabIndex = 19, ;
        Top = 180, ;
        Width = 144, ;
        Name = "cboP_mode2"


    ADD OBJECT lblp_mode2 AS label WITH ;
        AutoSize = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "PAY_MODE", ;
        Left = 186, ;
        Top = 187, ;
        Width = 64, ;
        TabIndex = 18, ;
        Name = "lblP_mode2"


    PROCEDURE cmdgrpeditor.cmdadd.Click
```

```
        *sele proj1
        APPEND BLANK

        modified = .f.
        Thisform.setall("Readonly",.F.,"Textbox")
        Thisform.setall("Readonly",.F.,"combobox")
        Thisform.setall("Readonly",.F.,"EDITbox")
        *thisform.txtyear.readonly = .T.
        thisform.cmdgrpnavigator.enabled = .F.

        thisform.cmdgrpeditor.cmdadd.visible = .F.
        thisform.cmdgrpeditor.cmdmodify.visible = .F.
        thisform.cmdgrpeditor.cmdrevert.visible = .T.
        thisform.cmdgrpeditor.cmdsave.visible = .T.
        thisform.cmdgrpeditor.cmdsave.enabled = .T.

        *
        THISFORM.REFRESH()
    ENDPROC


    PROCEDURE cmdgrpeditor.cmdsave.Click
                replace who with thisform.txtwho.value
                replace when with thisform.txtwhen.value
          replace where with thisform.txtwhere.value
                replace how with thisform.cbohow.value
                replace whenfirst with thisform.txtwhenfirst.value

                replace related with thisform.chkrelated.value
                replace periodic with thisform.chkperiodic.value
                replace prob_type with thisform.cboprob_type.value
                replace prob_desc with thisform.edtprob_desc.value
          replace complaint with thisform.edtcomplaint.value

          replace components with
thisform.pageframe1.page1.edtcomponents.value
          replace observation with
thisform.pageframe1.page1.edtobservation.value
          replace siteaction with thisform.pageframe1.page1.edtsiteaction.value

          replace date_in with thisform.pageframe1.page2.txtdate_in.value
          replace date_out with thisform.pageframe1.page2.txtdate_out.value
          replace shopaction with
thisform.pageframe1.page2.edtshopaction.value
          replace result with thisform.pageframe1.page2.edtresult.value
          replace returned with thisform.pageframe1.page2.chkreturned.value

        thisform.cmdgrpeditor.cmdadd.visible = .T.
        thisform.cmdgrpeditor.cmdrevert.visible = .F.

        thisform.cmdgrpeditor.cmdmodify.visible = .T.

        thisform.cmdgrpnavigator.enabled = .T.
        Thisform.setall("Readonly",.T.,"Textbox")
        Thisform.setall("Readonly",.T.,"EDITbox")
    ENDPROC
```

```
PROCEDURE cmdgrpeditor.cmddelete.Click
        store 0 to rep1
        *rep1 = messagebox("Are you really sure ?", 36, "Want to Delete ?")
        *if rep1 = 6 then
        DELETE
        PACK
        THISFORM.REFRESH()
        *endif
        if this.parent.cmdadd.visible == .F.
                this.parent.cmdadd.visible = .T.
                this.parent.cmdrevert.visible = .F.
                this.parent.cmdsave.enabled = .F.
        endif
ENDPROC


PROCEDURE cmdgrpeditor.cmdexit.Click
        THISFORM.RELEASE()
ENDPROC


PROCEDURE cmdgrpeditor.cmdrevert.Click
        if modified == .f.
        GO BOTTOM
        DELETE
        PACK
        else
                        replace who with mwho
                        replace when with mwhen
             replace where with mwhere
                        replace how with mhow
                        replace whenfirst with mwhenfirst

                        replace related with mrelated
                        replace periodic with mperiodic
                        replace prob_type with mprob_type
                        replace prob_desc with mprob_desc
             replace complaint with mcomplaint

             replace components with mcomponents
             replace observation with mobservation
             replace siteaction with msiteaction

             replace date_in with mdate_in
             replace date_out with mdate_out
             replace shopaction with mshopaction
             replace result with mresult
             replace returned with mreturned

        endif

        Thisform.setall("Readonly",.T.,"Textbox")
        Thisform.setall("Readonly",.T.,"EDITbox")
        thisform.cmdgrpeditor.cmdadd.visible = .T.
        thisform.cmdgrpeditor.cmdmodify.visible = .T.
        thisform.cmdgrpeditor.cmdrevert.visible = .F.
        thisform.cmdgrpeditor.cmdsave.enabled = .F.


        thisform.cmdgrpnavigator.enabled = .T.

        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpeditor.cmdmodify.Click

        Thisform.setall("Readonly",.F.,"Textbox")
        Thisform.setall("Readonly",.F.,"editbox")

        thisform.cmdgrpnavigator.enabled = .F.

        modified = .t.

                        Mwho = thisform.txtwho.value
                        mwhen = thisform.txtwhen.value
              mwhere = thisform.txtwhere.value
              mhow = thisform.cbohow.value
                        mwhenfirst = thisform.txtwhenfirst.value

                        mrelated = thisform.chkrelated.value
                        mperiodic = thisform.chkperiodic.value
                        mprob_type = thisform.cboprob_type.value
                        mprob_desc = thisform.edtprob_desc.value
              mcomplaint = thisform.edtcomplaint.value

              mcomponents = thisform.pageframe1.page1.edtcomponents.value
              mobservation = thisform.pageframe1.page1.edtobservation.value
              msiteaction = thisform.pageframe1.page1.edtsiteaction.value

              mdate_in = thisform.pageframe1.page2.txtdate_in.value
              mdate_out = thisform.pageframe1.page2.txtdate_out.value
              mshopaction = thisform.pageframe1.page2.edtshopaction.value
              mresult = thisform.pageframe1.page2.edtresult.value
              mreturned = thisform.pageframe1.page2.chkreturned.value

        thisform.cmdgrpeditor.cmdadd.visible = .F.
        thisform.cmdgrpeditor.cmdmodify.visible = .F.
        thisform.cmdgrpeditor.cmdrevert.visible = .T.
        thisform.cmdgrpeditor.cmdsave.visible = .T.
        thisform.cmdgrpeditor.cmdsave.enabled = .T.


        *
        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdtop.Click
```

```
        GOTO TOP
        THISFORM.REFRESH()


        this.parent.cmdprevious.enabled = .F.
        this.enabled = .F.

        this.parent.cmdnext.enabled = .T.
        this.parent.cmdbottom.enabled = .T.
ENDPROC


PROCEDURE cmdgrpnavigator.cmdprevious.Click
        IF !BOF()
                SKIP -1
                this.parent.cmdbottom.enabled = .T.
                this.parent.cmdnext.enabled = .T.
                IF BOF()
                        GO TOP
                        this.parent.cmdtop.enabled = .F.
                        this.enabled = .F.

                        this.parent.cmdnext.enabled = .T.
                        this.parent.cmdbottom.enabled = .T.
                else

                        this.parent.cmdtop.enabled = .T.
                        this.enabled = .T.
                ENDIF
        ENDIF
        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdnext.Click
        IF !EOF()
                SKIP
                this.parent.cmdtop.enabled = .T.
                this.parent.cmdprevious.enabled = .T.
                IF EOF()
                        GO BOTTOM
                        this.parent.cmdbottom.enabled = .F.
                        this.enabled = .F.

                        this.parent.cmdprevious.enabled = .T.
                        this.parent.cmdtop.enabled = .T.
                ENDIF
        ENDIF
        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdnext.MouseDown
        LPARAMETERS nButton, nShift, nXCoord, nYCoord
ENDPROC
```

```
	PROCEDURE cmdgrpnavigator.cmdbottom.Click
		GO BOTTOM
		THISFORM.REFRESH()


		this.parent.cmdnext.enabled = .F.
		this.enabled = .F.

		this.parent.cmdtop.enabled = .T.
		this.parent.cmdprevious.enabled = .T.
	ENDPROC


ENDDEFINE
*
*-- EndDefine: form1
**************************************************************************


PUBLIC oform1

oform1=NEWOBJECT("form1")
oform1.Show
RETURN


	**************************************************
*-- Form:         form1 (c:\nknew\ftrouble.scx)
*-- ParentClass:  form
*-- BaseClass:    form
*
DEFINE CLASS form1 AS form


	Top = 20
	Left = 65
	Height = 316
	Width = 437
	DoCreate = .T.
	Caption = "PC MAINTENANCE AND  TROUBLESHOOTING "
	Name = "Form1"


	ADD OBJECT lbltopic AS label WITH ;
		AutoSize = .T., ;
		FontSize = 14, ;
		FontUnderline = .F., ;
		WordWrap = .T., ;
		BackStyle = 0, ;
		Caption = "GEN. TROUBLESHOOTING RULES                & PROCESS",
;
		Height = 46, ;
		Left = 48, ;
		Top = 12, ;
		Width = 315, ;
		TabIndex = 1, ;
```

```
                ForeColor = RGB(255,0,0), ;
                Name = "lblTopic"


        ADD OBJECT edtdesc AS editbox WITH ;
                Comment = "", ;
                Height = 240, ;
                Left = 38, ;
                TabIndex = 3, ;
                Top = 58, ;
                Width = 360, ;
                ForeColor = RGB(0,128,0), ;
                ControlSource = "tchap21.desc", ;
                Name = "edtDesc"


ENDDEFINE
*
*-- EndDefine: form1
**************************************************
```