FIG. **6** illustrates another structure of dual mode application program.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Attention is initially directed to FIG. **1**, which depicts the block diagram structure of a prior art personal computer. The computer **100** comprises of interfacing circuits **101** and internal components **102**. Interfacing circuits are defined by the components provided to interface between the internal components of the computer with the user or the external environment. Typical interfacing circuits includes the display unit **111**, a screen to display information for the user; the audio circuit **112** for providing audio output; the keyboard **113** for entering alpha numeric information; pointing devices **114** represented by mouse, track ball and touch pad sensors. The ports **115** include all different kinds of communication interfaces such as USB port, fire wire port, serial and parallel ports, network port and modem. The internal components **102** of a computer comprise an internal hard drive **121** provided for storing OS, programs and data. Drives **122** are provided to accept external storage media. Typical external media accepted by the drives **122** are magnetic diskettes and optical disks. Some add on flash memory modules are marketed as solid state drives. In addition to the internal hard drive, the computer requires faster internal solid state memory **123** for providing higher speed operation. BIOS is represented by battery back up memory or flash memory storing the instruction set to be run upon power up or resetting the computer. The core of the computer is a high speed processor **125** which is sometime named as a CPU. When the power of the computer is turned on, the processor **125** fetches instructions from the BIOS **124** which setup the primary configuration of the computer. Then the computer is directed to initiate programs stored inside the hard drive **121**. The operation system, also commonly referred as the OS stored in the hard drive **121** further set up the hardware and software configurations of the computer. After the OS is booted, application programs stored inside the hard drive **121** can be launched to service the needs of the users. In order to take over the control of the computer just in case a crash occurs, the BIOS **124** may instruct the computer to look at the external drive **122** for initial program before proceeding to launch the OS from the hard drive **121**.

FIG. **2** illustrates the communication ports provided to a typical computer. Serial port **211** and parallel port **212** enable the computer to communicate with external devices such as printers. USB port **213** is another universal serial communication interface that provides limited power to the external devices. New computers may also provide interface connector **214** for connection with flash memory modules such as compact flash, smart media or memory stick. Network port **215** and modem port **216** enable the computer to communicate with external network. PCMCIA slot **217** is a standard interface port providing power and high speed interface for notebook computers to connect with miniature slim size devices such as network card or miniature drives. Some desk top computer may provide extra hard drive slot **218** connecting an external hard drive to the computer. IR port **219** and RF port **220** are provided for the computer to communicate with external devices without a cable. Blue tooth and WiFi are common RF standards used for note book computer.

FIG. **3** demonstrates the booting sequence of a typical computer. Programs instructions stored inside the BIOS memory **301** are executed by the CPU when the power of the computer is turned on. After setting up the primitive configurations of the computer such as the video display and the keyboard, the computer starts to boot the operation system **302** which is also referred as the OS of the computer. When Windows or Apple OS are booted, the OS provides a desktop, which enables user to access different application program **304** by clicking the short cut icons displayed on the desktop screen.

Attention is now drawn to FIG. **4**, which illustrates the flow chart for a computer to access an application program **431** after executing the instructions stored in the BIOS memory **401**. Since the BIOS **401** only provides very primitive setup of the computer, this pre-OS environment provides limited features for the pre-OS application program **431**. Accordingly the application program may include a primitive but improved pre-OS for enriching the support of the application program **431**. The mechanism to initiate the path **422** can be achieved by modifying the content of the BIOS, or by providing another initialization program stored on a diskette, which directs the computer to set up the secondary OS or to launch the application program **431** before booting the primary OS **402**. When a user is working on the application program **431**, the computer may continue to boot the primary OS **402** at the background. After the primary OS is booted, the computer launches the desktop and the full-featured application program **404** for replacing the primitive pre-OS version of the application program **431**. It should be noted that the path **423** and **421** may be provided as an option because many users may only want to have a brief access to the application program **431** before turning off the computer.

The primitive pre-OS version of the application program **431** and the full featured application program may be derived from two different programs, or from a packaged software providing services in both the pre and post OS environments. The pre-OS application program may be designed to provide some major functions of the full-featured application program **404**. In order for the application program **431** to be smoothly transformed over to **404**, a dual mode application program comprising both the application programs **431** and **404** is design to provide a common data file **505** as illustrated in FIG. **5**. Pre-OS application program represented by the block **501** of FIGS. **5** and **431** of FIG. **4** is configured to access the data file **505**. After the post OS full feature application program **502** of FIG. **5** or **404** of FIG. **4** takes over the control, the full features application program may also obtain access to the common data file **505**.

Due to technical constrains, sometimes it is preferable to provide different data files to each of the pre-OS and primary OS application program as illustrated in FIG. **6**. For example the data file **605** managed by the pre-OS application program **601** provides only the basic data information for the user. The data file **606** may provide additional imaged data information not available from the file **605**, for accessing by the post OS application program. Although the data file **605** and **606** are designed basically to store the similar type of information, the data file **606** may provide extended information about the application less desirable to be accessed by the pre-OS application program **601**. An example of the extended information to be stored in the file **606** but not in the file **605** are graphical information of large file size. In order to maintain the consistency of data retrieved in both the pre-OS and primary OS modes, the data files of the dual mode application program is to be synchronized as shown in the step **607** when the operation is switching from one mode to another.

Another reason to support the structure of FIG. **6** is that sometimes both the pre-OS application program and the corresponding application data file are stored in an external memory means. Since these devices provide limited memory