TurboDOS™
A Multiprocessor Operating System

By Mr. William A. Schultz
President, MuSYS Corp.

The TurboDOS operating system is a product of Software 2000, Inc., and is trademarked and copyrighted by them. At MuSYS Corp., we have used TurboDOS in conjunction with various slave processor boards to construct a wide variety of S-100 based computer systems, ranging from two to over sixty users.

TurboDOS is designed for multiprocessor networks of Z-80 based computers, although single user versions are available. Extensive use is made of the Z-80 instruction set to achieve a highly table oriented and reentrant architecture, which is very adaptable to the user's environment. In addition to MuSYS, many companies are selling TurboDOS for specific hardware configurations on an OEM basis. This is one of the primary distinctions with other multiprocessor operating systems, which are supported by only a single vendor.

While TurboDOS is currently restricted to Z-80 processors, a preliminary version of an 8086 slave processor is working, and it is expected that this will be released sometime late this year. The full 8086 implementation will be written in the "C" language, and will be available sometime in 1985. Eventual conversion to other processors will depend on the availability of software tools for porting TurboDOS onto any given CPU. This version of TurboDOS is expected to successfully compete against UNIX as well as the traditional CP/M-MP/M versions for 16 bit machines. TurboDOS will offer interface layers to application software written for some other operating systems, including the MSDOS/PCDOS family.

The multiprocessor nature of TurboDOS is it's most unique feature. Unlike other operating systems where networking of processors is a recent add-on, or which really only support a file transfer protocol, TurboDOS was designed from the ground up as a multiprocessor operating system. This means that the application program does not have to be aware of the nature of the hardware configuration it is running on. It can run in a single user system, a network master (Server), or a network slave (Requestor), with no modification. The user has a consistent operating method in all of these same environments. He does not have to pay attention to loading his networking package, nor does he have to modify the application software. In all cases, TurboDOS allows each user the private use of single user software and data bases, and also allows complete sharing of files, if the application has been written

with multiuser shared data bases in mind. In effect, a TurboDOS network system can be viewed as a single computer, in which each user has access to any resource in the system, if allowed by his privilege level and configuration.

Since TurboDOS is designed as a CP/M compatible operating system, well over 95% of CP/M software will run without any modification. Most of the software which does not run properly malfunctions due to direct BIOS calls for disk I/O, which cannot be successfully emulated in a multiprocessor network of any kind. As software vendors become more aware of TurboDOS, the percentage of compatible products will increase, and the number of vendors supporting extra functions for use under TurboDOS will also increase.

Also, since TurboDOS is a multiprocessing network based operating system, each user executing an application package appears to have the dedicated attention to the entire system, with very little degredation due to the operations performed by other users of the system. While 2-3 users can kill a bank switched system (e.g., WordStar, under MP/M), MuSYS has systems in the field running 12 users at a time in an intensive software development and execution environment for educational purposes.

Major Features of TurboDOS

In addition to the items discussed above, TurboDOS has several major features which should be mentioned.

TurboDOS supports very large files (up to 134MB) and disk drives (up to 1048MB). However, many programs which were only written for CP/M will not recognize files larger than the older CP/M limits of 512KB or 8MB. The largest disks in use on a regular basis are about 150MB, since single disks larger than that become cost prohibitive. With the continuing advance of technology, the TurboDOS limits will be approached.

TurboDOS is modular in construction, with operating system generation based on a relocating, linking, loader program. This makes the incorporation of different hardware driver modules quite easy, particularly for bus oriented systems, such as the IEEE-696 (S-100) bus commonly used for TurboDOS systems.

User friendliness is a key attribute of TurboDOS, with meaningful diagnostic messages, real alternatives for error recovery, a consistent set of command line switches for utility programs, and many small touches that give a minicomputer feel to this microcomputer operating system.

TurboDOS Throughput Considerations

One of the main benefits of TurboDOS is better throughput than standard CP/M. However, performance comparison

are very dependent on the specific implementations being compared and the applications being executed.

The networking aspect of TurboDOS has both positive and negative impacts on performance. Any individual application will run somewhat slower under networking than in a single user environment due to the overhead of the network. However, in the multiprocessor systems for which TurboDOS is designed, several users, each executing in a separate processor, will achieve far greater throughput than could ever be obtained in a single processor system, even if the single processor is a relatively advanced 16 bit system.

TurboDOS is based on a sophisticated buffer manager which performs multiple buffering of disk I/O. It uses a least recently used algorithm, with a slight bias toward directory blocks. This significantly reduces the number of physical disk access required, particulary as the number of users on the system increases. Typically, one processor in the network is devoted solely to buffering disk I/O to and from a hard disk, providing 16-24K of disk buffers, in addition to control of the network and print despooling for one or more printers. Use of this processor for program execution is usually restricted to system functions on a semi-dedicated basis, such as when disk format or back-up is required.

The operating system kernel of TurboDOS is fully reentrant and allows disk operations for several users to proceed on a time shared basis. This provides a significant increase in performance when multiple disk controllers are used.

Having said all of the above, the primary determinent of system throughput is the sector size and interleave factor used for the various disks on the system. This is as true for other operating systems as well, and it is perhaps a little unfair to compare CP/M, with a standard sector size of 128 bytes and an interleave factor of 6 (6 revolutions to read one track of data), and TurboDOS, with a standard sector size of 1024 bytes and an interleave factor of 1 (1 revolution to read one track of data). It is primarily on this basis, in conjunction with the buffer manager mentioned above, that claims for greatly increased speed of TurboDOS over CP/M are made. Based on the characteristics and criteria for the selected benchmark, a good CP/M implementation will perform on a par with the TurboDOS system, perhaps winning, or perhaps losing. However, good CP/M implementations are very rare. Good TurboDOS implementations are the rule, rather than the exception.

Networking Considerations

Since TurboDOS is primarily a network operating system, a few words on possible networking configurations are in order. TurboDOS does not impose an arbitrary network architecture on the system designer. There is no bias toward a particular Ring, Star, or Heirarchical network. The designer is left to his own devices, which are both good and bad points.

The primary consideration for a network interface is speed, since most users are concerned about throughput. If a TurboDOS network interface is being used, as opposed to a simple file transfer program, it can be assumed that the amount of data being transferred is very large. The TurboDOS network protocol imposes an overhead of between 50 and 100% on top of the actual data being transferred. This includes such things as packet headers, register snapshots, FCB snapshots, and other internal information. These days, most people complain about the slowness of floppy disks, and they operate at 250-500K BPS. Most hard disks operate in the range of 5000-10000K BPS, or about 20 times

faster than a floppy disk. Most of the serial networking hardware available today is in the range of 20-1000K BPS, with the majority of offerings at the top end of that range. Even there, the speed is only about twice as fast as a floppy disk, and a user who is used to hard disk operation will complain about that speed. For these reasons, MuSYS is trying to promote the use of ETHERNET, which runs at 10000K BPS, as the default standard for interprocessor networking. Recently, low cost ETHERNET chip sets and board level products have become available at the retail level. Eventually, ETHERNET will become a standard, primarily due to the market power of XEROX, INTEL, and DEC, who combined efforts on the design. We have demonstrated an ETHERNET implementation for TurboDOS which showed almost no degregation when accessing the remote hard disk, as compared with accessing the local hard disk. That compares with serial networks that crawl along at painfully slow rates.

The TurboDOS network is packet oriented, with a 16 bit address field. This allows up to 65,536 discrete processors, each supporting one or more time shared tasks. When a task in a processor accesses a remote resource, TurboDOS builds the packet and passes it to the network circuit driver, who is totally responsible for seeing to it that the packet gets to where it is addressed. Replies return in a similar manner. For this reason, it makes no difference if the network is point-to-point or a multiple access type. This choice can be made on the basis of cost, expected traffic patterns, and desired throughput, without consideration of any fixed requirements of TurboDOS.

Due to internal constraints, such as table sizes and CP/M compatibility considerations, each processor is limited to a maximum of 16 disk drives, 16 print queues, and 16 printers, each of which may be either local (directly attached to the processor) or remote (attached to another specified processor in the network). Certain additional functions, primarily those associated with communications channel I/O, may also be either local or remote to a designated processor. However, the total number of disks, print queues, and printers in a given network is effectively unlimited, since there is no requirement that every user have access to every peripheral. In fact, most implementors of larger networks will severely limit the global resources available to all users, and restrict the majority of accesses to the relatively local resources of an individual mainframe (which still could be a multiple processor system). An example of this would be a network of departmental level systems, with one globally accessible disk used as a central mailbox repository. This central system could also have other global resources, such as a high speed line printer, magnetic tape drive, or typesetting equipment. Anything which is too expensive to buy more than one of is a good candidate for the central system.

Conclusions

Each of us needs to rethink our own purchasing priorities. The cost of computer hardware is going down, and the cost of good software remains high. The usefulness of TurboDOS is that it provides access to a wide range of high quality software, both in CP/M compatible and enhanced shared operation, with high throughput at a low cost per user station. The really significant problems do not lend themselves to containment on the small floppy disk drives found on most "workstation" computers. With a TurboDOS network, each user can have access to one or more large hard disks, with all of the throughput and capacity advantages that entails, and still have a much lower cost per user than any discrete computer approach.