036

WORDPLAY OCTOBER 1987

8710

The PUNN Newsletter-Portland, Oregon

What's Inside

Club Officers

President	Dale Kirkwood 646-4354
Vice-President	(open)
Secretary	Don Barker 223-1749
Treasurer	Mike King 357-4413

PUNN Staff

Librarians	Ron Mayer 232-7363
	Walt Morey 239-5105
	Jim Thomas 284-2425
Hardcopy	Mike Calkins 636-1839
Program Chairman	Ted Peterson 244-1587
Workshop Chairman	(open)
Membership Chairman	Terry Priest 649-9583
Newsletter Editor	Charles Ball 639-0466

16576 SW Matador Lane-King City, OR 97224

BBS Committee

Chairman:	Al Kinney 640-5860
	Ron Mayer 232-7363
	Mike King 357-4413
BBS Phone Number	503/233-6804

From the Editor

On March 10, 1876 Thomas Watson, an assistant to Alexander Graham Bell, heard the words "Mr. Watson, come here. I want you." It seems that Bell was working in another room and had spilled some acid on his clothes and Watson was in another room. Thus the telephone was born and I wonder what those pioneers would think of their invention today?

Today in our computer world we now use this invention in a manner never contemplated by those early scientists. A modem that attaches to your computer and will open the "window of the world." Still we rely on the fundamental principal of that original telephone in order to communicate from one computer to another.

In our club we still have some members who for one reason or another do not use a modem. Whatever their reason or your reason for not entering this fascinating phase of computing, you can now put aside cost and the fear of not knowing how. Modems can be had for as little as $60.00, even less for a used one and our club has many experts that can show you how to use this marvellous device.

Once you get underway using your modem you will wonder why it took you so long to enter this area of computing. You will be able to use our PUNN BBS to exchange messages and download all your favorite programs. Besides that you will find many other BBS's that offer programs and services of every kind imaginable.

You're really not using your computer to its fullest ability if you're not into communications.

News & Views

By the time you read this many members will have travelled to Seattle to take in the TI Faire that was held on Saturday September 26th - - -We will report on all the things that were seen in the November issue of WordPlay - - -Dan Hawes fell off of his bicycle while going to work the other day and pretty badly scratched up his face-We understand he is now back in school and at work - - -Mike King reports the treasury shows a balance of $1368.43 as of September 1 - - -Ron Mayer has agreed to serve as Chairman of the nominating committee-he will be assisted by Chuck Neal and Al Kinney - - -Ted Peterson plans to continue his program on Word Processing at the October meeting-plan on attending this very interesting phase of the computer world - - -Chuck Ball will also show how he puts the newsletter together at this meeting-good information for you 'would be editors' - - -Inside this issue you will find a review of the 9640 Geneve by Peter Hoddie of the Boston Computer Society-you'll also find lots of other information important to your computer system - - -Your editor is constantly looking for programs and information that the members need-why not send something in?-you'll not only be helping your fellow members but you will get to see your name in print!

9640, Enlightenment

by J. Peter Hoddie, Boston Computer Society

I would like to set the story straight on hardware compatibility with the 9640. First of all, the TI, Cor-comp, and Myarc disk controllers will all work. It doesn't matter which eproa you have in the card. The TI controller can handle 80 track drives (just not in double density), the Cor-comp controller and the Myarc controller can handle 80 track and 16 or 18 sectors per track. The new reason for this is that the EPROM or ROM in the disk controller is not used by the 9640, but is replaced with code in the operating system. This allows the TI and Cor-comp controllers to run as fast as the Myarc currently does. The speed of disk access is really impressive - you may not recognize your disk drives. Any RS232 card from TI, Myarc or Cor-comp will work. Print spooling is built into the system for all cards, and the size of the spooler can now be set by the user. The print spooler is accessed just like a normal drive, such as PIO, rather than SPPIO as on the Myarc 512 card. The Horizon Ram disk will work, however, at this time in order to boot te system from it, it must use the HORIZON EPROM from Genial Computerware. This is not a ploy for me to make lots of money, but a decision made because of several unfortunate characteristics of the ROS distributed with the Horizon card. Currently there is support for only one Horizon Ram Disk, although this could change in the future. The Myarc 512 card can not be used as it is. However, for $15.00 Myarc will convert it so that it can be used as additional memory for the 9640. Once this change is made, the 512 card can not be used with the /4A, so carefully consider having this modification made. The speech synthesizer is supported but you have to buy a special card to put it into the expansion box. Such a card is available from Rave 99 for about $40.00. Your TI 32k or other memory cards such as Foundation will not work. Since the 9640 has over 600K of memory in its minimal configuration, this should not prove any grat hardship. At this time, the Megatronics GRAM card is not supported. The Cor-comp triple tech card will work, except that because of a somewhat faulty hardware decision (works on the /4A but not the 9640) the triple tech card will eat up about 1/8 of your available memory. The 9640 also supports an internal RAM disk which can be set to any size by the user, within the constraints of available memory. The current Myarc Winchester Personality card is supported, and of course the new Myarc hard drive/floppy controller will be supported when it becomes available. I hope this paragraph has clearcd up any misunderstandings you may have had about the 9640 and your present hardware setup. Please let me know if you have any further questions.

The documentation of the 9640 doesn't currently mention some of the more interesting features that are in the computer. For example, all disk files are available and date stamped at creation and at any update. This information is available on disk catalogs, and even from Basic using an extension of the current method of cataloging a disk. The RAM disk support is done similarly to the Myarc MPES (midi-peripheral expansion system), in that if you assign the internal RAM disk to drive 1, you can then make your physical drive 1 respond as drive 2. This means that all drives can be made always available, which is not always possible on the /4A. This is done independent of DSU base, thanks to the single master DSR (devise service routine) created for the 9640. For the assembly programmer there is a wealth of system utilities for graphics available through XOPs, written by Chris Faherty. The operating system also supports a new powerful set of disk access commands designed by Paul Charlton, and implemented by both of us. These allow for easy file and disk access from assembly for disk and file copying and comparing. The operating system also supports multi-tasking when not in /4A mode. This means you could be editing a file with your word processor, while down loading a file from a bulletin board, while a graphic image of a Frog dances on the corner of your screen. Multi-tasking allows you to run several programs at once - and this should open up some exciting possibilities in the future.

Until the operating system is released for the 9640, I would recommend taking anything you read from outside Myarc sources with a grain of salt. That is to say, without naming names, that I have read numerous articles on the 9640 which contain information that is just plain wrong. The articles claim that the machine can't do certain things, or that it will eventually do somethings better than it does now - and they are just completely wrong. While articles on the 9640 by people who have them at this stage are rather popular because people are crying out for any information they can get, many of those writing are very badly informed. This proble is as much a fault of Myarc as anyone. To release the hardware with incomplete software to anyone but developers was a serious mistake in my estimation. It has calmed many people down, but it has started a new furor over "where is the operating system" which is just as bad as the old "when will it be released". Lou Phillips has a habit of saying things to calm people down. If someone asks him when a product will be ready he tends to give the absolute best case answer. Unfortunately in this business, that tends to be way off base.

With this issue of WordPlay we are featuring HALLOWEEN! This festive event in October is something the kids have enjoyed for many years. We want you our readers to enjoy it too, so type the program in. It's easy and you too can enjoy this yearly event.

Some folks say that Halloween dates clear back to 700 AD. The word Halloween came about because it is celebrated on October 31, the day before 'All Saints Day'. The word Halloween is derived from hallowed or holy evening.

In early times in the United States Halloween was the time for the children to play harmless pranks, but in later years the harmless pranks were not so harmless. Now there seems to be less vandalism but instances of 'doctored' treats have worried many parents.

We want your Halloween to be safe and fun and that is why we publish this little program.

```
TYPE IN THIS PROGRAM SO THAT YOU CAN SHOW THE KIDS AND EVERY ONE ELSE A NICE LITTLE HALLOWEEN PROGRAM.
-------------------------------
100 !*HALLOWEEN* BY MIKE CHU *POMONA VALLEY USERS GROUP*
-------------------------------
110 CALL CLEAR :: CALL SCREEN(14):: CALL MAGNIFY(3)
-------------------------------
120 CALL CHAR(96,"010307020F0290B1621B0F0D06030100E0C0C080B0C0E0E0E0E0C0C0E01B1C0F")
-------------------------------
130 CALL CHAR(100,"000302060F1E7C7F4F0F0F0F0F070300C0E0B0F7E0703CFCECE4E0F0FBFEFF01")
-------------------------------
140 CALL CHAR(104,"301B0D071F3F7F7F9F9F0FFFF7B7B390F0060FBFCFEF7E3E3C17FFFF73606BCFB")
-------------------------------
150 CALL CHAR(114,"00010103FFFFFF03070FFFFFF070F0FF0F0E0F0FFFFFFC0C0C0FFFFFF80B0B0")
-------------------------------
160 DATA "HOW MANY JACK-O-LANTERNS ARE YOU GOING TO MAKE ?",1,4
-------------------------------
170 DATA "HOW MANY WITCHES WILL THERE BE ON HALLOWEEN NIGHT ?",2,10,"HOW MANY GHOSTS DO YOU THINK WILL COME OUT ON HALLOWEEN NIGHT ?",3,14
-------------------------------
180 READ A$,B,MX
-------------------------------
190 DISPLAY AT(21,1)ERASE ALL BEEP:A$ :: ACCEPT AT(24,1):N
-------------------------------
200 IF N>MX THEN A$="SORRY, TOO MANY TO FIT IN THE PICTURE ! - LESS THAN "&STR$(MX):: GOTO 190
-------------------------------
210 ON B GOTO 220,230,240
-------------------------------
220 FOR I=1 TO N :: CALL SPRITE(#I,104,10,144,INT(RND*248+8)):: NEXT I :: GOTO 180
-------------------------------
230 FOR I=5 TO N+4 :: CALL SPRITE(#I,96,2,INT(RND*88+8),INT(RND*248+8),0,-14):: NEXT I :: GOTO 180
-------------------------------
240 FOR I=15 TO N+14 :: CALL SPRITE(#I,100,15,184,INT(RND*248+8),-INT(RND*5+5),-2):: NEXT I
-------------------------------
250 F$="rtrtrtrtrtrt" :: G$="susususususu" :: DISPLAY AT(16,1)ERASE ALL:F$&" "&F$:G$&" "&G$
-------------------------------
260 DISPLAY AT(10,6):"HAPPY HALLOWEEN !" :: FOR S=1 TO 8 :: CALL COLOR(S,16,1):: CALL COLOR(9-S,2,1):: NEXT S :: GOTO 260
-------------------------------
```

Mass-Transfer, a review

PROGRAM NAME: MASS TRANSFER-Version 4
AUTHOR: Stuart Olsen
6625 W. Coolidge Street
Phoenix, Arizona 85033
REQUIRED: TI-99/4A console
32K Expansion-Disk Drive
RS-232 Card-Modem
XBasic or Ed/ASS
PRICE: $10.00 (Fairware)

Telecommunications is the name of the game, and what a game. But to make the game more fun and less frustrating requires a good modem-terminal emulator program. Mass-Transfer Version 4.2 fits that requirement.

In todays marketplace (commercial and fairware) there are several good to excellent modem programs to choose from. Each has its strong points and weak points. Mass-Transfer is one of the better programs.

The ease of use is by far the best of any program available for the TI today. This program can virtually be run without even printing out the rather complete set of instructions. Virtually everything is menu driven. All that is needed for most choices is one key stroke. No more remembering which combinations of keys to use to do the most basic of steps.

The main menu (which can be reached from the "terminal mode" by simply pressing FCTN 7 or AID) contains the following selection:

(R) Reconfigure modem port
(C) Clear download buffer
(U) Upload DIS/VAR 80 file
(A) Auto-dial from directory
(L) Linefeed toggle status=OFF
(E) Echo Remote status=OFF
Monitor status=ON
(B) Buffer capture status=ON
(S) Set up log file status=OFF
(H) Hangup after MXTstatus=OFF
(X) Modem file transfer
(M) Multiple Xmodem Transfer (MXT)
(V) View buffer contents
(D) Dump download buffer
(F) Files (catalog disk)

Those that require additional prompts, such as (R), (X), (A), and (S) are also all menu driven. A big help to us who can't keep track of the instructions. There are however, a few commands which involve the use of the Function key. They include F9 for a screen dump, F1 to change text color, and F= to turn the print spooler on and off.

One of the more unusual features of Mass-TRansfer is its ability to do "Ymodem" file transfers. The only catch there is that there isn't really a "standard" Ymodem protocol as of now. I don't know of any BBS that has that protocol available. So covering that procedure is of little value at this time.

Included in the documentation are complete instructions to modify the program to meet your particular needs. It works flawlessly with the Horizon Ram Disk. You can easily set the defaults to your baud rate, number of stop bits, screen color, print spooler type and more.

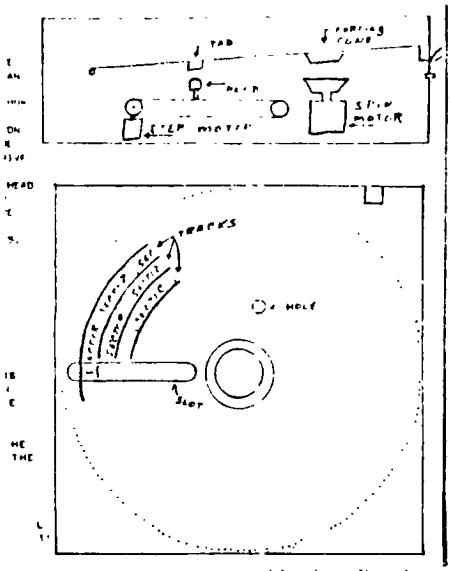
Probably the most useful feature is the auto-dialer. Included with the Mass-Transfer disk is a program PHONEMAKE. In it you set up your phone directory and modem commands. Up to nine different directories may be set up. When using the auto-dialer, you have the choice of using the redial feature. The redial feature will redial the desired number until the line is no longer busy or until you cancel it. This is an extremely useful feature.

Stuart Olsen has done a fine job with Mass-Transfer. If you use it send him the $10.00. This insures that as the program is upgraded you will be able to get the upgrades. This program is available from the PUNN library. -By Tom Wills

The Disk Drive System

If you want to have a full understanding of how you computer and the programs and files that it deals with work, you must also understand the disk drive system. If you could see how a disk drive worked with the cover off it would go a long way to help you understand the system.

The disk drive reads and writes on concentric circular tracks on the disk. The disk is clamped at the center hub and is spun by the drive motor inside the protective jacket at 300 rpm. When you close the drive door the disk is clamped to the drive motor. Also, the head and the pad lightly squeeze the disk at the slot in the cover of the disk. The head is mounted on a slide and connected to a step motor. This motor moves the head in and out in discrete steps. One step equals one track on the disk. There are normally 40 tracks on a disk. Each track is divided into 9 sectors. (18 for double den- sity) Each sector can hold 256 bytes. So that means that 40 tracks x 9 sectors (18 sectors) x 256 = 92,160 (184,320) bytes per disk side.



A new disk is blank. You have to initialize it before you can use it. This is necessary so that the disk operating system (DOS) can use the disk. DOS is one of those computer buzz words like MS DOS, PC DOS, etc. TI DOS is one of the best. More on that later. When you initialize a disk, you name it and set up the tracks. There is a special chip in the controller that picks read or write, the drive select, and which track and sector to go to. As you use the disk a rec- ord is kept of what is where and how much space is left on the disk. The first part of the disk has a directory on it and each track has information on it, too. It's somewhat like labeling your file cabinet, the drawers, and how many file folders are in each drawer. As you use it, you're labeling the type of information in each folder.

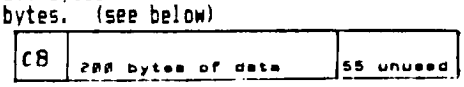
When you initialize a disk, you are putting on the disk information that your computer needs to use the disk. There is a built in routine to make use of the disk information. The name and programs on the disk are on sector 0. So how do we get to it? When we start the disk drive it lines up the hole and light shines through it. Then the head is moved all the way to the outside of the disk, and there is sector 0! This is where the name, # of sides, track density, and the number of free tracks remaining is kept.

TI DOS puts 9 sectors (18) into each of the 40 tracks with 256 bytes in a sector. IBM uses 9 sectors (double density) with 512 bytes for the same capacity. Seems the same, but with small groupings there is less waste with the TI system. If you use part of a 256 byte sector you waste less than using part of a 512 sector.

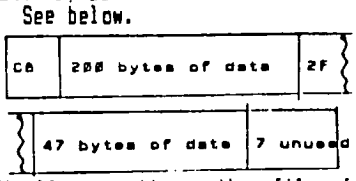
The disk directory will tell you the type and location of the files that are on the disk. Each file put on the disk has an entry on the directory. Typically, bytes 1-2 are the total number of sectors in the file. Byte 3 has file flags that tell the type of record (bit 1,0=fixed length 1= variable length). Bit 4 is the protect flag. Bit 6 "basic" file information-0=display format, 1=internal format. Bit 7 type of file 0= data 1=program image. Byte 4 number of rec- ords per sector. Byte 5 end of file offset. Byte 6 record size, for fixed length-it will equal actual length, but variable length will contain the maximum record size that is allowed. Bytes 7 8 total of records in the file.

There are three types of files used; 1-Fixed length record; 2- Variable length record; 3-Basic program image. Fixed length records are user specified, and it means all files will have the same length. If a file with 80 byte records has 3 records, then 240 bytes will be in the sector that is used to store it. The other 16 bytes will not be used.

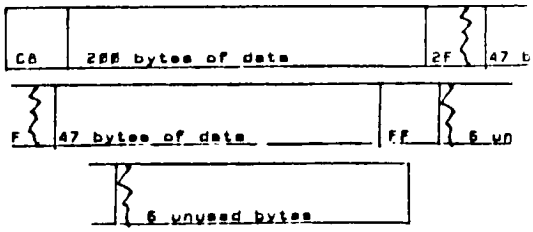
Variable length record files have a length that the user specifies up to the maximum allowed (254). 254 is the maximum allowed because the record length and end of file take up the other two. For example, we have a file of 200 bytes with a maximum record size of 250. >C8 (decimal 200) goes on first and then the 200 bytes of data. This leaves 55 unused bytes. (see below)



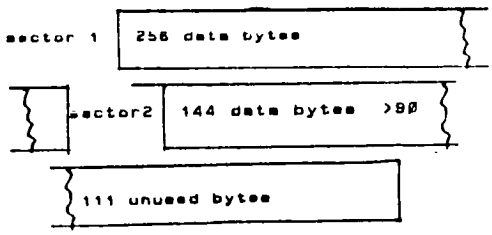
Next we write a file of 47 bytes and send it to the disk. >2f (decimal) 47 is written and then the 47 bytes of data. Now 249 bytes of the sector have been used. See below.



Finally we write another file of 50 bytes. Needing 51 bytes to write it to the disk, we cannot get it on this sector. An end of file mark is written on the sector and the new file is written on the next sector, and we leave 6 bytes unused. See below.



Basic program image files start with the first sector and fills each sector with program data until all the data is saved. The last sector of the file may now be completely full. For example, we have a 400 byte long program and we save it to disk. Sector one will be full (256 bytes), and the second sector will have 144 bytes. There will be a pointer byte right after the 144 bytes containing >90 (144 decimal)



To get the most out of this article, you should have a program that allows you to read a disk sector by sector. (Advanced Diagnostics is such a disk, but there are others.) There are 40 tracks with 9 sectors on a track. This is a total of 360 sectors, yet when you put a new disk in and initialize it, up pops 358 sectors for use! There are even less for the user to actually use. Sectors 0 to 21 are reserved for the operating system.

The TI disk is divided into blocks called allocatable units (au's). An au is equal to one sector of 256 bytes. That is 4096 for DS/DD. AU #0 makes up the volumn information block (VIB) that has the disk name, number of AU's, number of sectors per track, number of sides, and the allocation bit map. AU #1 has an alphabetical index of all the files on the disk. The rest of the AU's have the file descriptor blocks and data. File descriptor blocks are like the VIB, but they refer to the files.

Bytes 0-9 contains the disk name. The name can be any combination of ten ASCII characters except for a space or period.

Bytes 10-11 This gives the total number of AU's on the disk.

Byte 12 Indicates the number of sectors per track.

Bytes 13-15 Contains the ASCII characters for "DSK". The disk manager looks to see if these are present. If not, it assumes an uninitialized disk.

Byte 16 contains the ASCII code for "P" if the disk is protected. If not, this will be a space character.

Byte 17 Indicates the number of tracks per side.

Byte 18 Shows how many sides have been formatted.

The Disk Drive System

Byte 19 Indicates the density of the disk.
Bytes 20-55 are reserved for future use. Set to zero.
Byte 56-255 contain the allocation bit map. These 200 bytes can keep track of up to 1600 256-byte records, or about 400k-enough to handle double sided/double density formatting. Each bit represents a sector. If a sector is in use the bit is set to one, if not it is set to zero.

```
   +-------------------------+
 0 !                         ! 1
   =      D I S K  N A M E    =
 8 !                         ! 9
   +-------------------------+
10 !   TOTAL NUMBER OF AUs   ! 11
   +-------------------------+
12 !#SECT/TRACK !    "D"      ! 13
   +-------------------------+
14 !    "S"     !    "K"      ! 15
   +-------------------------+
16 ! PROTECTION !#TRACKS/SIDE! 17
   +-------------------------+
18 ! # OF SIDES !   DENSITY   ! 19
   +-------------------------+
20 !                         ! 21
   = RESERVED FOR FUTURE USE =
54 !                         ! 55
   +-------------------------+
56 !                         ! 57
   =    ALLOCATION BIT MAP   =
254!                         ! 256
   +-------------------------+
```

Any changes are recorded in sector 0. This allows the DOS to locate the data. A file stored on disk is referenced by a file descriptor record. This tells the TI DOS what sector the file is stored at, if it is a program or a data file, and if the file is stored in one block or several noncontigous blocks. The FDR's are located on tracks 2-34 and are entered in the order they are created.

TI DOS uses sector 1 as an alphabetical index of all the file names currently on the disk. This is the file descriptor index re- cord (FDIR). The index consists of sector numbers. Each number refers to the FDR for that file. When a new file is created, the FDR's are scanned, sorted, and then their sector numbers are reprinted onto sector 1 in the new alphabetical order. This indexing helps speed up file access and cut down on wasted disk space.

FDR DESCRIPTION

Bytes 0-9 contain the filename (up to ten ASCII characters).

Bytes 10-11 Reserved for future expansion.

Byte 12 This is the file type flag. The bits are set according to the file attributes in TI Basic and can be interpreted as follows:

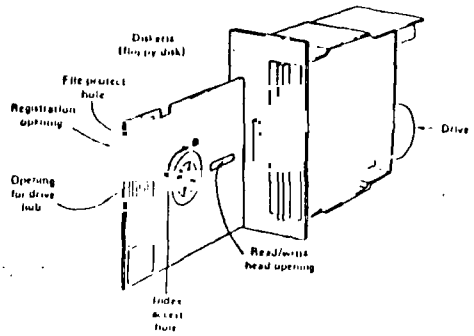
BIT	MEANING
0	-0=Data File 1=Program
1	-0=Display 1=Internal
2	-Reserved future use
3	-0=Un-Pro'td 1=Pro'td
4-6	-Reserved future use
7	-Fixed L'gth 1=Var. L'gth

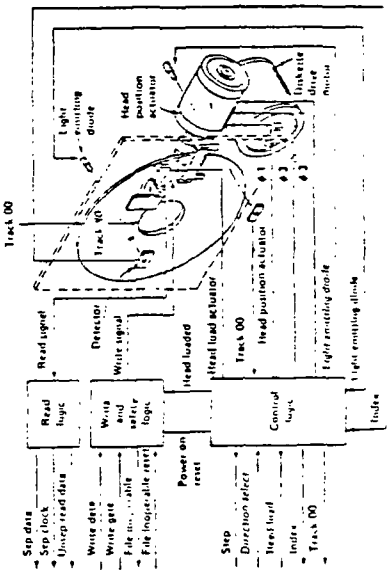
Byte 13 contains the number of sectors used by the file.
Byte 14-15 contain the number of sectors used by the file.
Byte 16 contains the end of file offset. This value is used to locate the last byte in the file. This prevents reading past the end of the file. It is used for variable length and for program files.
Byte 17 contains the record length. If variable length, it will be the value for the maximum allowed.
Byte 18-19 contain the number of records allocated for the file. Either the number currently on the file or the number of files it was "opened" for in TI Basic. If a variable type file this value will be the same as in Bytes 14-15 but in reverse order.
Bytes 20-27 Reserved, set to 0.
Bytes 28-255 contain the data pointers. When the file must be broken up due to size, a reference to the next record of the file is entered into the pointer area. This tells TI DOS where on the disk to find the next block of records for this file. Each data chain pointer is made of two sections of three bytes. The first entry is the sector number of the start of a new data block. The second entry is "EOF OFFSET" of that block-this is not necessarily the EOF of that file. To make things worse, the three bytes are stored in a strange way (see note below).

```
   +++++++++++++++++++++++++++
 0 +                         + 1
   ~      F I L E  N A M E    ~
 8 +                         + 9
   +-------------------------+
10 +        RESERVED         + 11
   +-------------------------+
12 + FILE TYPE !RECS PER SECT+ 13
   +-------------------------+
14 +   # OF SECTORS USED     + 15
   +-------------------------+
16 + EOF OFFSET ! REC LENGTH + 17
   +-------------------------+
18 +   # OF RECORDS USED     + 19
   +-------------------------+
20 +                         + 21
   ~        RESERVED         ~
26 +                         + 27
   +-------------------------+
28 +                         + 29
   ~   DATA BLOCK POINTERS   ~
254+                         + 255
   +++++++++++++++++++++++++++
```

(NOTE): The bytes are stored in reverse. Then the six byte segment is stored as shown. As each new block is created, a six byte entry is added to the data chain pointer area. The pointer area can handle up to 76 different blocks for the same file.

Now get out your disk reading program and start looking for the various bits of information. Note the volumn information and the file information records. After a bit of exploration and experience, you should be able to recover blown disks, deleted files and other disk problems.






```
**********************************
*                                *
*  Murphy's Rule:                *
*                                *
*     Every job will take twice  *
*     as long as originally plan-*
*     ned and be half as lucrative.*
*                                *
**********************************
```

Net Worth

This program comes to us as freeware via Innovative Programing. The Author is John Galen and if you take a moment and type it in you may discover some interesting things about yourself.

Every one should have all of their financial affairs in order. This includes the facts and figures of their assets and liabilities written down. The time may come when you quickly may need to know just exactly what you are worth and with this handy program you can keep accurate records.

You might need to establish a loan for a house, a car or some other important need. Well if you keep this program up-to-date you'll know immediately where you are and this could go a long way in convincing a lender that your financial affairs are in order.

The program provides a procedure to list all your assets and liabilities and to establish the values attached to each item. It then will automatically determine your true net worth after you have made all the entries. As your financial condition changes you can provide the new figures to the program and an up-dated status will be given. It even includes a print out if you have a printer and you can file this data for review from time to time.

Many of us are rather lax on keeping the facts and figures of our financial affairs current. If you use this program it can help you in many ways. You could use it for substantiating insurance claims. If you should be audited by the IRS, (heaven forbid) it could be an assist it establishing certain financial data. The main thing is that if you have not already established a good method of keeping track of all your financial data, this program may be just for you.

You might even be surprised to find out just how much you are worth when you sit down for a minute and type in your assets and liabilities. Put in every thing that you own that has a value and put in everything that you owe on. If you already own a house it's worth a little more each month. If you are paying on a car, you owe a little less each month. Don't forget cash values of things such as life insurance policies. You may have certain benefits from your employment that would contribute to your net worth. Take a moment some evening and work with this program. Who knows? You may be rich beyond your wildest dreams.

Try Blinking

When you're working on your video display terminal, BLINK, and you'll eliminate one cause of eye fatigue, says an Ohio opthalmologist. "While using a VDT," wrote Dr. Frank J. Weinstock in a letter to the Journal of the American Medical Association, "the user has a tendency to stare and decrease blinking to avoid missing anything on the screen."

This, and not the screen itself, is what usually causes a sense of eye fatigue, he wrote. Blinking provides eye lubrication and reduces that 'tired-eyes' feeling significantly, he said.

Could This be You?

I intended to publish this program earlier this summer, but then I thought the better of it. I think that now that summer has just about come to an end that you should type it in

It's a short program but it is I believe very descriptive of some of our members. It may even make some of you feel rather guilty about some of the opportunities you passed up this past summer. Now it is too late and the opportunities afforded by this program will just have to wait until next year.

(The editor)

```
100 REM **A GLIMPSE OF REALI
TY
110 REM     FOR COMPUTER ADI
CTS
120 CALL CLEAR
130 CALL CHAR(96,"1818183C3C
7E3C18")
140 CALL CHAR(112,"FFFFFFFFF
FFFFFFF")
150 CALL CHAR(120,"FFFFFFFFF
FFFFFFF")
160 CALL CHAR(121,"555555555
5555555")
170 CALL CHAR(122,"5D5D5D5D5
D5D5D5D")
180 CALL CHAR(128,"0000011925
438FF5D")
190 H=22
200 CALL COLOR(9,16,1)
210 CALL COLOR(11,2,2)
220 CALL COLOR(12,13,1)
230 CALL COLOR(13,14,1)
240 CALL VCHAR(4,16,112,3)
250 CALL VCHAR(4,17,112,3)
260 CALL VCHAR(4,20,112,18)
270 CALL HCHAR(22,1,120,96)
280 CALL HCHAR(3,18,112)
290 CALL HCHAR(2,17,112,3)
300 G=0
310 H=H-1
320 FOR Z=7 TO H-1
330 CALL VCHAR(Z,16,96)
340 CALL VCHAR(Z,16,32)
350 NEXT Z
360 G=G+1
370 CALL SOUND(15,(H*150),2)
380 CALL HCHAR(H,G,121)
390 IF G=32 THEN 300
400 IF H=7 THEN 410 ELSE 320
410 FOR F=3 TO 30 STEP 3
420 CALL HCHAR(7,F,128)
430 CALL VCHAR(8,F,122,14)
440 CALL SOUND(30,(F*200),2)
450 NEXT F
460 PRINT "NOW DO SOMETHING
ABOUT IT!!"
470 GOTO 470
```

Big Letters

Have you ever needed some REAL BIG letters? If you have such a need type in this program. It will display on your screen any character that you type in and it fills the complete screen with just that one character!

If you have a screen dump program you can print the large character out on your printer. I tested the program and printed out the characters using Quality Software "Screen Dump II" and a load interupt switch.

This is another program from Jim Peterson.

(Editor)

```
100 DIM X$(96):: CALL CLEAR
:: FOR CH=33 TO 89 STEP 8 ::
 FOR A=0 TO 7 ! REAL BIG LET
TERS by Jim Peterson
110 CALL CHARPAT(CH+A,X$(CH+
A-32)):: CALL CHAR(CH+A,"0")
:: L$=L$&RPT$(CHR$(CH+A),3):
: NEXT A
120 FOR T=1 TO 3 :: R=R+1 ::
 DISPLAY AT(R,4):L$ :: NEXT
T :: L$="" :: NEXT CH
130 CH$(1)=RPT$("0",16):: CH
$(2)=RPT$("F",16)
140 CALL SOUND(100,500,0)
150 CALL KEY(0,CH,S):: IF S=
0 OR CH>96 THEN 150
160 CALL HEX$(X$(CH-32),B
$):: FOR J=9 TO 64 :: CALL C
HAR(J+32,CH$(VAL(SEG$(B$,J,1
))+1))
170 NEXT J :: GOTO 140
180 SUB HEX$(H$,B$):: HX$
="0123456789ABCDEF" :: BN$="
0000":: FOR X=1 TO LEN(H$)
190 FOR J=LEN(H$) TO 1 STEP -
1 :: X$=SEG$(H$,J,1)
200 X=POS(HX$,X$,1)-1 :: T$=
SEG$(BN$,X*5+1,4)&T$ :: NEXT
 J :: B$=T$ :: T$="" :: SUBE
ND
```

Yesterday is a memory - Tomorrow is a vision - Today is a bitch!

Program for Net Worth

```
100 ! NET WORTH PROGRAM
110 ! WRITTEN BY:
120 ! GALEN READ
130 ! AUTHOR OF WriterEASE
140 !           99'er
150 !           CONSOLE CALC
160 !           MEMORY PLUS
170 !           MFI
180 !           NETWORTH
190 !           CHARACTERS
200 !           EASY SPELLER
210 !           DISK DOCTOR
220 !           AND MORE!
230 ! INNOVATIVE PROGRAMMING
240 ! P.O. BOX 2737
250 ! ROHNERT PARK
260 ! CA. 94928
270 ! (707) 585-3922
280 ON WARNING NEXT :: ON BREAK NEXT
290 LOD$="PIO" :: OD$="DSK1.DATA" :: DIM A$(500),L$(500),A(500),L(500)
300 CALL CHAR(132,"0000003F202F2F2C2D0000FC04F4F434B4B434F4F404FC0000F2D2C2F2F203F")
310 CALL CHAR(140,"FF00FFFFF0FF3000B4B4B4B4B4B4B4B40000FFF00FFFF00FF2D2D2D2D2D2D2D2D")
320 CALL CHAR(137,"FCFCDC9C00F9CDCFC0000B0B83E1C0B003F3F3B3900393B3F")
330 CALL CLEAR :: CALL SCREEN(2):: FOR X=0 TO 14 :: CALL COLOR(X,16,5):: NEXT X
340 CALL B1 :: DISPLAY AT(2,6):"NET WORTH PROGRAM": :" INNOVATIVE PROGRAMMING"
350 DISPLAY AT(7,2):"PRESS:" : :"1 TO ADD or CHANGE ASSET": :"2  ADD or CHANGE LIABILITY": :"3  DISPLAY NET WORTH"
360 DISPLAY AT(15,1):"4  PRINT HARDCOPY": :"5  LOAD or SAVE DATA": :"6  END PROGRAM"
370 DISPLAY AT(7,1):"PRESS:" :: FOR X=0 TO 9 :: CALL KEY(0,K,S):: IF S=1 THEN 210
380 NEXT X :: DISPLAY AT(7,1):: FOR X=0 TO 9 :: CALL KEY(0,K,S):: IF S=1 THEN 210
390 NEXT X :: GOTO 180
400 IF K=54 THEN 640 ELSE IF (K<49)+(K>53)THEN 430 ELSE CALL B2 :: ON K-48 GOTO 220,300,380,430,520
410 ! ADD OR CHANGE ASSET
420 DISPLAY AT(10,2):CHR$(138)&" NAME OF ASSET "&CHR$(138): : : : :"PRESS ";CHR$(139);"ENTER";CHR$(137);" FOR MENU"
430 CALL HCHAR(11,2,139):: CALL HCHAR(11,31,137)
440 ACCEPT AT(11,1)VALIDATE(UALPHA,DIGIT):AST$ :: IF AST$="" THEN 150 ELSE DISPLAY AT(15,1):"LOOKING FOR ASSET"
450 FOR X=0 TO AX :: IF AST$=A$(X)THEN 270
460 NEXT X :: DISPLAY AT(15,1):"THIS IS A NEW ASSET" :: A$(AX)=AST$ :: X=AX :: AX=AX+1 :: GOTO 280
470 DISPLAY AT(15,1):"PRESS ";CHR$(139);"ENTER"&CHR$(137);" TO KEEP VALUE"
480 CALL HCHAR(11,2,143):: CALL HCHAR(11,31,141):: CALL HCHAR(13,30,137)
490 DISPLAY AT(13,1)SIZE(-27):"ENTER VALUE "&CHR$(139);STR$(A(X)):: T1=T1-A(X):: ACCEPT AT(13,14)VALIDATE(DIGIT,".")SIZE(-14):A(X):: T1=T1+A(X):: GOTO 220
500 DISPLAY AT(10,2):CHR$(138)&" NAME OF LIABILITY "&CHR$(138): : : : :"PRESS ";CHR$(139);"ENTER";CHR$(137);" FOR MENU"
510 CALL HCHAR(11,2,139):: CALL HCHAR(11,31,137)
520 ACCEPT AT(11,1)VALIDATE(UALPHA,DIGIT):LBL$ :: IF LBL$="" THEN 150 ELSE DISPLAY AT(15,1):"LOOKING FOR LIABILITY"
530 FOR X=0 TO LX :: IF LBL$=L$(X)THEN 270
540 NEXT X :: DISPLAY AT(15,1):"NEW LIABILITY" :: L$(LX)=LBL$ :: X=LX :: LX=LX+1 :: GOTO 360
550 DISPLAY AT(15,1):"PRESS ";CHR$(139);"ENTER"&CHR$(137);" TO KEEP OLD VALUE"
560 CALL HCHAR(11,2,143):: CALL HCHAR(11,31,141):: CALL HCHAR(13,30,137)
570 DISPLAY AT(13,1)SIZE(-16):"ENTER VALUE "&CHR$(139);STR$(L(X)):: LT=LT-L(X):: ACCEPT AT(13,14)VALIDATE(DIGIT,".")SIZE(-14):L(X):: LT=LT+L(X):: GOTO 300
580 DISPLAY AT(7,1):"NET WORTH IS" :: DISPLAY AT(10,1):USING 590:T1,LT,ABS(T1-LT)
590 IMAGE TOTAL ASSETS  #######.## TOTAL LIABILITY  #######.##  --------- NET WORTH..... ######.##
600 DISPLAY AT(15,1):"PRESS ANY KEY TO CONTINUE"
610 IF T1-LT=0 THEN DISPLAY AT(7,14):"NONE" ELSE IF T1-LT<0 THEN DISPLAY AT(7,14):"NEGATIVE" ELSE DISPLAY AT(7,14):"POSITIVE"
620 CALL KEY(0,K,S):: IF S=0 THEN 620 ELSE 150
630 DISPLAY AT(11,1):" ENTER OUTPUT DEVICE:":" ";CHR$(139);LOD$;TAB(21);CHR$(137);: ACCEPT AT(12,3)SIZE(-18):LOD$ :: DISPLAY AT(20,1)
640 IF LOD$="" THEN 150 ELSE ON ERROR 500 :: OPEN #1:LOD$ :: PRINT #1:" ASSETS:";TAB(32);"AMOUNT  LIABILITIES:";TAB(73);"AMOUNT"
650 FOR X=0 TO MAX(LX,AX)-1 :: PRINT #1:A$(X);TAB(30);:: PRINT #1,USING 660:A(X);:: PRINT #1:TAB(41);L$(X);TAB(71);:: PRINT #1,USING 660:L(X)
660 IMAGE #######.##
670 NEXT X :: PRINT #1:RPT$("-",80);"ASSET TOTALS";TAB(30);:: PRINT #1,USING 660:T1;:: PRINT #1:TAB(41);"LIABILITY TOTALS";TAB(71);:: PRINT #1,USING 660:LT
680 CLOSE #1 :: GOTO 150
690 CALL KEY(0,K,S):: IF S=0 THEN 690 ELSE 150
700 ON ERROR 510 :: CLOSE #1
710 DISPLAY AT(20,1):"OUTPUT DEVICE ERROR" :: CALL ERR(M,N,O,P):: GOTO 430
720 DISPLAY AT(11,2):"1 LOAD DATA": :" 2 SAVE DATA": :" 3 RETURN TO MENU"
730 CALL KEY(0,K,S):: IF S=0 THEN 730 ELSE IF K=51 THEN 150 ELSE IF K<>49 AND K<>50 THEN 730
740 CALL HCHAR(11+((K-49)*2),16,137):: CALL HCHAR(11+((K-49)*2),6,139):: DISPLAY AT(15,1):"DEVICE NAME";CHR$(139);OD$;TAB(28);CHR$(137)
750 ACCEPT AT(15,13)SIZE(-15):OD$ :: IF OD$="" THEN 150
760 ON ERROR 610 :: OPEN #1:OD$ :: IF K=49 THEN 790
770 PRINT #1:STR$(AX):: FOR X=0 TO AX-1 :: PRINT #1:A$(X):: NEXT X
780 PRINT #1:STR$(LX):: FOR X=0 TO LX-1 :: PRINT #1:L$(X):: PRINT #1:L(X):: NEXT X :: CLOSE #1 :: GOTO 150
790 T1,LT=0 :: INPUT #1:AX$ :: AX=VAL(AX$):: FOR X=0 TO AX-1 :: INPUT #1:A$(X),A(X):: T1=T1+A(X):: NEXT X
800 INPUT #1:LX$ :: LX=VAL(LX$):: FOR X=0 TO LX-1 :: INPUT #1:L$(X),L(X):: LT=LT+L(X):: NEXT X :: CLOSE #1 :: GOTO 150
810 ON ERROR 820 :: CLOSE #1
820 IF K=49 THEN DISPLAY AT(20,1):"INPUT DEVICE ERROR" ELSE DISPLAY AT(20,1):"OUTPUT DEVICE ERROR"
830 GOTO 540
840 CALL CLEAR :: CALL B1 :: DISPLAY AT(11,1):"LEAVING... ": :"  ARE YOU SURE ( Y / N )" :: CALL SCREEN(9)
850 CALL KEY(0,K,S):: IF S=0 THEN 850 ELSE IF K>96 THEN K=K-32
860 IF K=78 THEN 140 ELSE IF K<>89 THEN 850 ELSE CALL CLEAR :: STOP
870 GOTO 870
880 SUB B1
890 CALL CLEAR :: CALL HCHAR(1,1,132):: CALL HCHAR(1,2,142,30):: CALL HCHAR(1,32,133):: CALL HCHAR(24,2,140,30)
900 CALL HCHAR(24,32,134):: CALL VCHAR(2,1,143,22):: CALL HCHAR(24,1,135):: CALL VCHAR(2,32,141,22)
910 SUBEND
920 SUB B2
930 DISPLAY AT(9,1):RPT$(CHR$(142),28):: : : : : :RPT$(CHR$(140),28): : : :
940 CALL HCHAR(9,2,132):: CALL VCHAR(10,2,143,6):: CALL HCHAR(9,31,133):: CALL VCHAR(10,31,141,6):: CALL HCHAR(16,2,135)
950 SUBEND
```

Fast Resigns

At the regular board meeting on Tuesday September 15th the Board members were presented with the following letter.

"Due to increasing demands on my time, I do regretfully submit my resignation as president of Portland User's of ninety-nines, effective this date September 15, 1987.
I do leave this position thankful for the chance to serve with such an enthusiastic group of people. I will continue to remain as involved as time permits, and look forward to the groups continued growth."

(signed) Kieth Fast

Because of this resignation Dale Kirkwood, vice-president will fill the unexpired term of president until our new election this coming December. An interim vice-president will be appointed.

P.O. Box 15037
Portland, OR 97215

DISCLAIMERS: The PUNN User's *Group is not affiliated* with or sponsored by TI and has no relationship with them, implied or otherwise.
Mention of a company or product is not an endorsement of that company or product.
We are not a subsidiary or branch of any other User's Group and any relationship we may have with other groups is on the basis of equals.

ALL GENERAL MEETINGS ARE HELD ON THE FIRST TUESDAY OF EACH MONTH, AT THE PGE BUILDING 3700 SE 17TH. PORTLAND, OREGON

!! NEXT MEETING DATE !!
OCTOBER 6TH. 1987

THE **PUNN** NEWSLETTER

Wordplay

P.O.BOX 15037
PORTLAND, OREGON 97215

POINTING THE WAY FOR
USERS OF TI's 99/4 COMPUTERS

OCT 1987-VOLUMN VI ISSUE 10