

# THE APPLE /// SYSTEM CONFIGURATION PROGRAM (SCP)

By Joe Dombrowski  
Apple Users Group International

Published by APPLE /// USERS OF NORTHERN CALIFORNIA (ATUNC Newsletter)  
VOL. 2 NO. 11 NOVEMBER, 1985

One of the most important things any /// user MUST understand is how to use the System Configuration Program (SCP) and how to build a SOS.DRIVER. Only you, the user, know your system's configuration and how you'll be using any piece of software. With this in mind we bring our first tutorial, one on SOS.DRIVER building.

## RAM, ROM AND TURNING IT ON:

Between the time you turn your computer on and the time you can do something useful with it, an awful lot of things happen - if everything goes well. This discussion concerns what should, or might, go on.

The first thing that occurs when you /// is turned on is that it checks its own hardware based on the instructions (diagnostics) in the Read Only Memory (ROM) chip. If any of these tests are failed you get one of the error messages listed on pg 113-114 of the Owner's Guide. These indicate very serious problems which will require servicing, so quit right there if you get one.

If these tests are passed, the machine then looks to the internal disk drive to read its next set of orders (booting). More exactly, it is looking for a file called SOS.KERNAL which contains the fundamental instructions the machine needs to function internally. Without these instructions the /// can do nothing.

Next the /// looks to the same drive for the file SOS.INTERP (interpreter) which contains the instructions for interfacing with you, the user.

Finally, it looks on the disk in the internal drive for a file called SOS.DRIVER which contains the instructions for the machine to manage all its other devices such as the console, printers, hard disk, etc. This file also must be present and must contain at least the CONSOLE.DRIVER. As you may have learned, the absence of any of these files prevents a successful boot and you get an error message telling you what is missing. These errors (pg 114-115), for the most part, do not indicate anything wrong with your machine, rather there is usually something wrong with the disk you attempted to boot and the message tries to tell you what.

While the 3 SOS files must be there, you do have some control over 2 of them. (The one you can't control is SOS.KERNEL which is the same for all boot disks.) You actually select the Interpreter. This file is an assembly language program which contains the language you are using such as BASIC or PASCAL, or the actual application program you are using such as Visicalc or Applewriter.

Hence you choose the interpreter when you choose the program. This also means that SOS.INTERP files are not freely interchangeable. The BASIC interpreter will not allow you to run Pascal programs even though it will boot. If you are building a boot disk, be sure you have copied the correct interpreter for what you want to do.

The one you can really manipulate to your advantage is SOS.DRIVER and you don't even need to know a thing about programming. SOS.DRIVER is actually the directory name of the file which contains the individual device drivers. A device driver is just an assembly language program that tells the computer how to read from and/or write to the device.

Using the System Configuration Program (which is only a fancy file mover) on the Utilities disk you can add device drivers to the file as you add devices. YOU CAN ALSO DELETE THEM. Why would you delete drivers? Usable

memory. When you boot your system you read all these individual drivers in SOS.DRIVER into your RAM memory. (See table of contents for list of drivers and their approximate memory requirements in K bytes.)

Thus SOS.DRIVER with several drivers could take up 40K or more of memory. That is 40K you cannot use for a VisiCalc template or an Applewriter document; and, particularly if you have only 128K, you will probably miss it sometimes. The secret is, AFTER MAKING A COMPLETE BACKUP COPY OF THE COMPLETE SOS.DRIVER FILE, to reconfigure the system using only those drivers you really need.

If you don't have a QUME or Silentype printer you don't need those drivers and pick up 8K. You only need .FMTDX on your Utilities disk (Backup ///, 3 EZ pieces etc.) since it is used only for formatting. Very few application packages use .GRAFIX, etc. A word of caution: Read the configuration section of an application program carefully to determine the drivers it requires.

For example, VisiCalc Adv Ver requires an .AUDIO driver. Still, the SOS.DRIVER that comes with it needs about 20K. If you use it with just a second floppy drive and a printer, you can cut that to about 14K by deleting .PROFILE and .QUME.

If you are wondering about the disk drive driver, don't. It is included in SOS.KERNAL because every /// has a disk drive. Admittedly there are some more things that may automatically happen when you boot a disk. The interpreter program, e.g. VisiCalc, may automatically start. Also, your HELLO (BASIC) or System.Startup (Pascal) program may automatically start (Turnkey disk) or you may get a message telling you to insert another disk (Two-stage boot). However, these events are not necessary for an Apple /// to get started. The sequence of SOS.KERNAL, SOS.INTERP and SOS.DRIVER is necessary and under those exact file names on the same boot disk. This is true even if you are using a "Hard Disk Boot" such as Catalyst or if you have PMOVED Pascal to Profile.

#### WHAT HAPPENS WHEN YOU BOOT YOUR DISK?

When you boot up an Apple /// application, the system loads the Operating System (SOS.KERNAL), the Interpreter or application (SOS.INTERP), and the Drivers (SOS.DRIVER) into memory from the top down. In downloading the drivers the system WILL NOT SPLIT A DRIVER BETWEEN TWO 32K REGIONS OF MEMORY. If the driver to be loaded will not fit in the current 32K bank the loader skips the rest of that bank and loads into the next lower 32K segment. That skipped space is NOT available to any application and is wasted.

With larger configurations of drivers like print spoolers, it becomes more likely that this will occur. With a 256K or 512K machine you can probably chalk this up to overhead, but on a 128K machine you will probably wish to optimize the situation if possible and experiment with progressively larger configurations.

The reason we can't give you firm guidelines on which drivers to load in what order is that it depends primarily upon factors which vary completely from system to system, user to user and application to application. See the next section for a review.

#### TECHNICAL NOTE ON SYSTEM LOADING OF DRIVERS

The three variable system parameters which affect the loading of any particular driver file, and thus the 32K (more or less depending on revision) boundary problem are:

- the size of the Interpreter loading below SOS but above the drivers.
- the number and size of each driver configured, and,
- the order of those drivers in your SOS.DRIVER file (which, by the way is the opposite of the order the SCP list them in).

One example should get you on the track.....

- start with a SOS.INTERP that is 25K long
- using SCP we build a SOS.DRIVER file by loading an existing file (normally a SOS.DRIVER, however in our case individual files) which contain .CONSOLE driver 5K and .PROFILE driver 3K.
- then add a 24K print spooler, let's use D.A. Datasystems .PPRINT (a parallel printer spooler) in our example.
- Since .PPRINT is added last it will be loaded before .CONSOLE or .PROFILE.
- Now save the new SOS.DRIVER on our new boot disk and boot it...

Let's discuss what this means.

-SOS.KERNAL loads from FFF to 8800 of the topmost reserved System Bank leaving 14K free in that bank (8000-B7FF). SOS.INTERP loads below that extending down 11K into the top User Bank, leaving 21K free in that bank. The 24K .PPRINT WILL NOT FIT in this remaining space so it is skipped and .PPRINT followed by the 8K of .CONSOLE and .PROFILE load into the next lower bank. The net result is 21K wasted space.

#### HOW TO MAXIMIZE MEMORY:

1. Select the minimum number of drivers necessary to utilize your application in the manner you wish.
2. Group drivers in near 32K groups (on a separate piece of paper).
3. Load driver, first the smallest group, then the larger (one nearest 32K-not over).
4. Edit driver if necessary.
5. Setting the System Parameters.
6. Generating the new system

(Originally published in the News and Views Newsletter, November 1985)