DOS/65 V3 SUPPLEMENT

VERSION 3.0A

© (Copyright) Richard A. Leary
2693 Haven Hill Drive
Grand Junction, CO 81506

richardaleary@gmail.com

.

TABLE OF CONTENTS

SECTION 1 - INTRODUCTION

This document is a brief supplement to the DOS/65 V2.1 documentation describing the changes associated with DOS/65 V3.0. While the changes are small in number, they significantly improve the utility of DOS/65 especially for those users having some sort of high capacity mass storage device attached to their system.

The key changes are:

- PEM and CCM changes to implement a CP/M 2.2 compatible “USER” area concept for storage devices.
- Addition of a UCOPY transient that is able to set source and destination USER areas for any file copy operation.
- CCM changes to implement a CP/M compatible batch command processing capability.
- Addition of the CP/M compatible SUBMIT transient command to initiate batch command processing by CCM
- Change to SYSGEN to allow either .KIM or .HEX input files for the BOOT and SIM modules used when building a new system
- Use of the IOSTAT defined but previously unused storage location as a means of exchanging USER and DRIVE data between CCM and SIM.

To implement these changes CCM has increased in length by one page to 2560 bytes. PEM length has remained unchanged at 3072 bytes.

All software has used 6502 opcodes and addressing modes. No 65C02-only code has been used.

The SYSGEN executable needed to configure and save a V3 system is SYSGN3nn.COM. That can be assembled from the supplied .ASM file and MAKECOM used to create the .COM file. Executing SYSGN3nn.COM then will integrate the user's BOOT and SIM files and save the resulting OS image to the user specified hard or floppy drive system tracks. All of these steps can be performed on a V2.17 lor later DOS/65 system.

As has been my policy for several years, complete source code is provided for all of the changed software, i. e., SYSGEN, UCOPY, and SUBMIT. I also provide an updated, but still very slow, version of FILESTAT.BAS (FILES3nn.BAS) that recognizes and shows the USER area information for files.

I have also provided my current BOOT, SIM, and MONITOR source code files as an additional source of information.

.

SECTION 2 – USER AREAS

2.1 OVERVIEW

DOS/65 V3 supports 16 user areas that enable hard drive (or solidstate equivalents such as CF or SD cards) storage organization more effectively than can be accomplished when the entire drive can only be viewed as a single large entity. These 16 user areas are identical to CP/M 2.2 user areas (sometimes called user codes).

2.2 STRUCTURE

User areas are implemented by using the first byte in each 32-byte directory entry as a location for storage of the user area associated with the file referenced by the rest of the directory entry. Access to any given file requires that PEM by instructed as to which USER area to search for or otherwise access a given file. This means that top-level commands issued by CCM under DOS/65 V3 will see or manipulate files only for the USER area specified to PEM.

CCM has been significantly modified with two visible changes.

First, a new built-in command (USER) has been added to CCM to set the system-wide user area. The syntax of this command is simple:

```
USER n
```

Where n is a number from 0 to 15.

Second, the CCM command prompt has been modified with the addition of a USER number. The CCM prompt now takes on the following form:

```
dn>
```

Where “d” is the drive letter (A to H) and n is the one digit or two digit USER area number (0 to 15).

PEM has also been significantly modified with one change visible at the PEM interface. A new function has been added with the following characteristics:

FUNCTION: X=36

> INPUT: New USER area in A using a binary 0 to 15. A=255 does not change the USER area but requests the current USER number from PEM.
> OUTPUT: Current USER number is returned in A when A at entry is 255.

It is important to note that normal operations will only affect files or use files in the USER area displayed in the CCM prompt.

SECTION 3 – BATCH PROCESSING

2.1 OVERVIEW

DOS/65 V3 provides a batch processing capability using files of type SUB that contain one or more lines that are identical to command lines that the user types in real time at the system console. It is the transient tool SUBMIT.COM that converts those lines saved in an ordinary text file into CCM command lines.

There is one restriction on use of SUBMIT in conjunction with the user generated SUB file. Both of those files must be on drive A for the SUBMIT-generated $$$.SUB file to be recognized by CCM and batch processing to be automatically initiated when SUBMIT executes a X=0 call to PEM after creation of the $$$.SUB file.

If SUBMIT.COM is run on a drive other than A the resulting $$$.SUB file will be created on the same drive (B through H). No automatic execution of the $$$.SUB file will take place. If the $$$.SUB file is moved to drive A and a warm boot is initiated then the commands in the $$$.SUB file will be executed.

Upon completion of execution of the commands in the $$$.SUB file CCM deletes the $$$.SUB file.

2.2 SUBMIT

SUBMIT.COM is a small transient that converts the users .SUB file into a file named $$$.SUB that has the commands presented to CCM in a way that is consistent with normal use of CCM by the user.

SUBMIT has two other powerful features:

First, SUBMIT provides a means of incorporating dummy parameters named $1 through $9 into the users SUB file and then replacing them with actual user-entered values when submit is run.
/
For example, assume that the user has created a file named MAKECOOL.SUB that contains the following lines:

```
asm cool
makecom cool
```

If the user then executes the following command

```
submit makecool
```

The lines in the MAKECOOL.SUB file will be executed and the final product will be the file COOL.COM.

However, SUBMIT has the ability to support real time parameter substitution so that a user created SUB file (MAKE.SUB) with the following content:

```
asm $1
makecom $1
```

When executed with the following command:

```
submit make cool
```

will accomplish the same end result as the MALECOOL.SUB but does it in a way that enables a single SUB file to be used with many different data files without having to create a new SUB file every time. With as many as nine parameters SUBMIT can support quite complicated parameter replacement in SUB files.

Second, SUBMIT can generate command lines that will contain control character inputs to CCM or the referenced transient. This is done simply by entering an up-arrow (ASCII $5E) followed by the associated alphabetic character. Submit changes those two characters in the SUB file to the actual control character in the $$$.SUB files.

The following command line shows an example of this feature:

```
help ^h
```

assuming that the transient HELP will do something in response to the control-h that SUBMUT will insert in the associated $$$.SUB file.

SECTION 4 - UCOPY

UCOPY.COM allows a file to be copied from any location to another location. It is similar to the standard transient COPY.COM in that the command syntax is:

```
UCOPY <from ufn> <to ufn>
```

The difference is that after the transient starts up it will ask for the user area number of the <from ufn> and separately ask for the user area number of the <to ufn>. UCOPY will copy the specified file. Since UCOPY users can specify both the from and to user areas a file can copies from anywhere to anywhere.

SECTION 5 IOSTAT

Maintenance and exchange of default USER and DRIVE data between the various DOS/65 V3 modules (SIM, CCM, and PEM) is a more difficult task than the default DRIVE only concerns of prior versions. Use of the defined but unused IOSTAT byte at $106 was a simple and logical choice as the likelihood of such an approach creating conflicts was minimal.

IOSTAT contents under V3 are now define as follows:

> BITS 0-2 – default DRIVE (0 to 7 corresponding to A to H)
> BIT 3 – unused
> BITS 4-7 – default USER (0 to 15)

Each module has specific responsibilities regarding this approach as described in the following paragraphs. The user need only ensure that their SIM complies with the stated needs as all other responsibilities are satisfied by the V3 code:

> SIM COLD BOOT – initialize IOSTAT. Normally this would mean setting DRIVE and USER to zero but there may be circumstances where the user prefers a different configuration.
>
> SIM WARM BOOT – use the DRIVE field in IOSTAT to set and select the default DRIVE.
>
> CCM – use IOSTAT upon entry into CCM to determine and then set the default DRIVE and USER. Setting those conditions requires that CCM send the appropriate commands to PEM while maintaining that information within CCM.
>
> CCM – preserve default DRIVE and USER information in IOSTAT for use during a WARM BOOT or return from a transient execution.
>
> PEM – set and report DRIVE and USER data and manage the file system in accordance with that data. Save IOSTAT contents when performing a WARM BOOT, X=0.

While the WARM BOOT JMP at $100 still results in execution of a WARM BOOT by SIM it may not always reflect the default DRIVE and USER data in IOSTAT. It is recommended that transients that end execution with a WARM BOOT do so by executing a JSR or JMP to PEM with X=0. PEM will ensure the default data is saved before jumping to the SIM WARM BOOT entry.

SECTION 6 – FILES3.BAS

FILESTAT.BAS does a good job of displaying details of the files on a drive but is pathetically slow. DOS/65 V3 does nothing to fix that issue – only an assembly language version will fix that and I have not put that task on the priority list. However, testing of various transients under V3 revealed that FILESTAT.BAS was ignoring the first byte in the directory entry and hence would “see” directory entries in all USER areas.

However, since FILESTAT.BAS was not written to look at the first byte in the directory what it reported was in error if the same named file was in multiple USER areas. It was also incomplete as it did not report the USER area for any directory entry.

The fix was simple and has been incorporated into FILESTAT V3 (FILES3.BAS). FILES3 sorts entries by USER and displays the USER area.

I completed an assembly language version. That is the now released SD (super directory).

Section 7 – BOOT – SIM - MONITOR

Examples of BOOT, SIM, and MONITOR are included in the distribution. The only significant functional changes are addition of IOSTAT initialization to the cold boot code in SIM-MONITOR and use of the drive field in IOSTAT during a warm boot. For further discussion of IOSTAT see section 5.

The other DOS/65 V3 SYSGEN change is that when executing SYSGEN the BOOT and SIM object code files may now be in either the KIM format generated by the DOS/65 assembler or may be in the HEX format generated by other tools, e. g., the WDC tools.

SYSGEN V3 (SYSGN3nn.COM) requires that the user enter the full name of the BOOT and SIM files. However, the TYPE field is not used to identify the type of file that has been specified. That is done automatically when SYSGEN V3 looks at the first valid character in the file. If that is a semicolon (;) then the file is processed as a KIM file and if that is a colon (:) then the file is processed as a HEX file.