System Manager's Guide

Technical Systems Consultants, Inc.

System Manager's Guide

COPYRIGHT INFORMATION

DISCLAIMER

The supplied software is intended for use only as described in this manual. Use of undocumented features or parameters may cause unpredictable results for which Technical Systems Consultants, Inc. cannot assume responsibility. Although every effort has been made to make the supplied software and its documentation as accurate and functional as possible, Technical Systems Consultants, Inc. will not assume responsibility for any damages incurred or generated by such material. Technical Systems Consultants, Inc. reserves the right to make changes in such material at any time without notice.

System Manager's Guide

I. The System Manager

Each UniFLEX™ (UniFLEX is a trademark of Technical Systems Consultants, Inc.) installation needs a system manager. This person is responsible for various system administration and maintenance. Large installations may require two managers, one for the administrative aspects, and the other for system supervision and maintenance. The system manager has a 'user id' on the system which allows any UniFLEX operation without question. He may delete other's files, change passwords, and so on, all at will. This 'power' should obviously not be abused.

The administrative duties of the system manager include creation of new users on the system, setting forgotten passwords, updating the 'message of the day' file, etc. The maintenance aspects include file system backups, system configuration, and file system repairs. Some of these will require a fair amount of skill on the part of the person in charge. Before reading this guide, you should become 'an expert' by reading (and understanding) the manual, 'Introduction to UniFLEX'. This document assumes the basic knowledge presented there. You should also read 'UniFLEX Utility Commands' since most of the commands referenced in this document are only mentioned and not fully explained.

II. Initial Setup

The system is supplied on 8" floppy disk and contains several pre-configured versions of the operating system. Separate sheets have been included which describe the various configurations (see 'Hardware Setup'). These have been named 'uniflex', 'uniflexl', etc., in the root directory. The initial system boot will use the standard configuration which is the file named 'uniflex'. After the system is up and running, as described below, select the configuration you need and simply rename that particular file to 'uniflex'. At boot time, the system loads and runs the file named 'uniflex' in the root directory. The sheets describing each version of UniFLEX on your disk also describe the hardware configuration for that version. Follow these exactly! If you don't, your system will not run! Note that these sheets describe 'standard' configurations. You should consult the 'System Configuration Guide' for custom system adaptation.

III. Starting the System

To start the system, the 'boot' must be executed. The boot program usually resides in a monitor ROM and is activated by typing 'D' in response to the monitor's prompt. You should consult the documentation from your hardware manufacturer for the specifics of booting. You must place the supplied floppy disk in disk drive 0. This is the only drive capable of booting the system as supplied.

The boot process takes several seconds, at which time the following should appear on the screen.

```
UniFLEX Operating System
Copyright (c) 1980 by
Technical Systems Consultants, Inc.

Version 1.00  Released September 2, 1980


Total user memory = 212K

++
```

The 'version' number and 'release' dates should be recorded in a convenient place. If you ever need to contact your manufacturer or Technical Systems Consultants, Inc., about the operating system, you will need this information, as well as the serial numbers from the computer and the software.

The message at the bottom specifies the available memory. This amount will vary depending on the particular configuration and the amount of memory installed in the computer. The last line output will be the system prompt '++'. UniFLEX is now in 'single-user' mode which means there is only one active terminal.

Single-user mode is normally used for system maintenance and general system setup. Several commands may only be run while in single-user mode such as 'format', 'flex', and 'stop'. The first time the system is brought up for the day, several operations should be performed as a matter of routine. The first of these is setting the date. The 'date' command is used for this and is done as follows:

```
++ date 9-2-80 7:30:22
```

The first entry is for the month, day, and year. The second is the time and should be entered as 24 hour time. The local time should be entered. Consult the 'System Configuration Guide' for details pertaining to 'time zone' configuration.

Another daily operation should be the editing of the 'message of the day' file. This file is named 'motd' and resides in the directory '/etc/log'. The file may be left empty or it may contain important information and news the system's users may need to know. As an example, if a new program is installed on the system, such as the Pascal compiler, you could create a message file as follows:

```
++ chd /etc/log
++ edit motd +b
#~d!
#i
     1.00=September 2, 1980:  A new Pascal compiler was
     2.00=installed on the system today.  Documents which
     3.00=describe its use are available in the main office.
     4.00=All users are welcome to use this new compiler.
     5.00=#s
++
```

The first line above will change your working directory to '/etc/log' which is where the motd file resides. The next line invokes the editor and tells the editor that no backup file is desired (the '+b' option). Next, the old contents of the file are deleted and the insert mode is entered. You may not always want to delete the old file's message, but instead, you may want to append to it. Note that the message includes the date it was created so that if several are appended, the date it was originally issued will be known. You should consult the 'UniFLEX Text Editor' manual for complete details on its use.

Another operation to perform while in single-user mode is device mounting. The system will boot up with a default root device, usually a fixed platter hard disk, but this depends on the configuration. Other disk devices may be needed while the system is running. As an example, let's assume the system boots with the hard disk 'hd0' as the root device (see the directory named '/dev' for the list of devices on your system). Let's also assume that we have a removable cartridge disk called 'rc0' we wish to access or have as part of the file system. The following will mount this device:

```
++ /etc/mount /dev/rc0 /usr2
```

The 'mount' command resides in the '/etc' directory and may only be executed by the system manager. The arguments listed above tell mount to take the device named 'rc0' and mount it on the node of the file system named '/usr2'. The '/usr2' directory is provided with most 'standard' systems and is normally kept empty for the purpose of mounting. When a device is mounted on a directory node, the contents of that directory become inaccessible for the duration of the mount. After the above mount operation, any references to the directory '/usr2' will actually access the root directory of the device 'rc0'. With this in mind, to access a file which is on the device 'rc0' called 'test', which is in a directory named 'exp' in the root directory of that device, the pathname:

```
     /usr2/exp/test
```

would be used. File systems (block devices) may be mounted anywhere in the directory tree structure.

Activation of printer spoolers should also be done while in single user mode. This is covered in section IX of this document. Another feature of UniFLEX is the 'history' file. The system will keep its own history (keep track of logins, boots, etc.) if a file named '/act/history' exists. To disable this feature, simply 'kill' this file. If enabled, the history file can become quite large. It is good practice to 'truncate' it about once a week. The command:

```
     ++ create /act/history
```

will truncate the file to zero length. The information it keeps is only useful for a short period of time, so keeping old history information on the disk is wasteful. See the 'history' command description in 'UniFLEX Utility Commands' for details of the history file's contents.

Since it is possible to create a file containing a list of commands, and then have 'shell' execute that file, you might want to create a file containing all of your usual system startup operations. It as also possible to create such a file and have UniFLEX execute them automatically each time the system goes from single-user mode to multi-user mode. The system always looks for a file named '/etc/startup' when entering multi-user mode. If this file exists, it is read as a list of command lines and executed. When this file is executed, there is no terminal associated with the running commands. If terminal input or output is desired, I/O redirection must be used on the command lines in the file.

Once all of the daily operations are out of the way, the system is ready for multi-user mode. To take the system out of single-user mode and into multi-user, simply type:

```
     ++ log
```

The system will delay for several seconds as miscellaneous housekeeping is performed. Finally, something like the following will appear on all configured terminals connected to the system:

```
     Technical Systems Consultants, Inc.
        UniFLEX Operating System

     07:33:12 Tue Sep 02 1980      tty00


     Login:
```

The system is now ready for user login, and any valid user name and password will allow system access.

IV. Shutting Down

At the end of the day (or scheduled session) the system will be brought down. This is a two step process. NOTE: NEVER RESET THE COMPUTER! Always follow this procedure when stopping the system. Any other method will more than likely destroy files on the system. The first step in shutting down is to use the 'shutup' command. This command may only be executed by the system manager. Normally you should run 'shutup' with an argument of 15 to 30 minutes. 15 is the default. Every five minutes or so after the invocation of 'shutup', a message will be sent to all terminals logged in to the system similar to the following:

```
The system will be down in 15 minutes.
```

This message cannot be locked out with the 'message' command so every user will be forewarned. At the end of the time period, the message

```
The system is now down.
```

will be sent to all users and their terminals will once again become inactive. The system console terminal (the terminal used when booting the system) will still be active and should respond with the prompt. If no prompt appears, just type a return and the prompt should show up. The system is once again in single-user mode and is in an identical state to that at boot time. You may decide to leave the system in single-user mode until it is time for the next multi-user session, or you may shut the system completely down by use of the 'stop' command. The 'stop' command will take the system from 'single-user' mode to the stopped state, and the message

```
System Stopped!
```

will be printed on your terminal. At this time the computer's reset may be used (this is the only time from the moment a boot is performed!).

V. Adding Users

One of the main responsibilities of the system manager is to maintain the user list. Anyone using the system will need a 'user name' and optionally a password. The 'user name' is usually the last name of the user or his initials. It is best to use names which actually identify the person to which they are assigned. All user names should meet the following requirements:

1. The name must be all lower case letters
2. The name must be 8 characters or less
3. The name must be unique

Strange results may occur if these rules are not followed. Passwords may be assigned by the system manager or the user may define his own. *It is wise to issue new users a password to stress their importance.* The user may always change his password at any time. Passwords can be almost anything but the following guidelines will make for a more secure system:

1. Passwords should be 5 - 6 characters in length
2. Make them a random mix of letters and numbers
3. Letters should always be lower case

Random strings are better than words or names. UniFLEX is supplied with two known user names, 'system' and 'bin'. Since neither of these have assigned passwords, you should assign them as soon as possible (see below).

A new user also requires a directory. This directory usually has the same name as the 'user name'. All user directories are normally placed in the directory '/usr'. The 'crdir' command is used to create the new directory. As an example, let's assume we have a new user named Mary Jones. The first step is to assign a user name and password. We will make the user name 'jones' and the password will be 'rizmox'. Next you should create the user directory as follows:

```
++ crdir /usr/jones
```

Before you can let the system know about the password you need to inform it about the new user. There is a file in the directory '/etc/log' called 'password'. This file is the list of all users, their user identifiers (user id for short), password, login directory name, and login program name. Each line of this file has the format:

```
name:password:id:directory:program
```

Each field is separated by a colon (':') and no spaces are permitted. The password field does not show the actual password, but shows an encrypted form of the password instead. Since anyone is generally permitted to read this file, the actual password is never used. The user id is a number which the operating system will associate with the user. Each user should have a unique user id. User id's should be assigned in a range from 1 to 32000. User id 0 represents the system manager (user name of 'system') and should never be assigned to anyone else! The login program entry is usually null in which case the system will run the 'shell' upon logging in. Continuing our example, you will need to edit the password file (/etc/log/password) and add the new user name to the end of the file. The line should look like this:

```
jones::164:/usr/jones:
```

Note that the password field is left null as well as the login program field. The user id of 164 was assigned to Ms. Jones. The next step is to change the owner of the new user's login directory. In this example, type:

```
++ owner jones /usr/jones
```

You must be logged in as the system manager before executing this command. Users should own their own directories so they may set the permissions on them as desired. The last step in installing this new user is to set the selected password. The following command will do just that:

```
++ password jones
New Password: rizmox
Please Retype: rizmox
```

The password will not be echoed when it is entered as shown above. The password command requests the password twice to make sure it is entered correctly.

As new users are added to the system, new terminals may be required. Adding terminals to the 'standard' configuration is quite simple. Once the appropriate hardware has been installed to support the terminal, the file '/etc/ttylist' needs to be modified. Examining this file will show several similar lines. Each line will look something like this:

```
+ tty00
  or
- tty06
```

The '+' tells UniFLEX to activate the specified terminal and the '-' causes it to be ignored. If you add a terminal, let's say it is 'tty06', simply edit the 'ttylist' file, changing the '-' to a '+' on the line for 'tty06'. Note that the names for the terminals may not be changed and the standard configuration will not support more terminals than listed in the supplied ttylist file. If a terminal is later removed from the system, simply set the '+' back to a '-' for the appropriate terminal.

VI. Adding New Programs

There is some choice as to where you should place system wide programs. Two system binary directories are searched by the 'shell' each time a command name is typed. The first of these is the '/bin' directory. The second is '/usr/bin'. It is a good idea to keep the number of files in directories to a reasonably small number. Once you go over 60 files in the '/bin' directory, you may see a slight decrease in system response. This is the primary reason for two system binary directories. The first one ('/bin') should be used for more often used programs, such as 'dir', 'shell', etc. If a program will be less popular, it should be placed in '/usr/bin'.

There is a third directory which is primarily for system only programs, such as 'format'. This is '/etc'. As programs are obtained which have restricted use or may only be executed by the system manager, they should be placed in '/etc'. Files appearing in this directory will not be automatically located by 'shell' so you must either be in the '/etc' directory at the time of execution, or specify '/etc' specifically in the command name.

VII. System Maintenance

Whether your system is supporting a user base of 2 or 200, your users will be expecting their files to be safe on the system. Unfortunately, it is ridiculous to expect a computer system to behave 100% of the time. There is bound to be an occasion when a power failure or power 'glitch' takes down the system, and probably will destroy files in the process. With this in mind, the system manager should make 'backups' on a daily basis. This is not a particularly time consuming task and should be made part of your daily routine. Backups may be made using the 'copy' command, or by making use of one of the available backup programs. The 'System Maintenance Guide' should be consulted for complete details. This guide also provides information for disk integrity checking, and for damaged file structure repair.

VIII. Making New Disks

All new disk media must be formatted by one of the available format programs before it can be used on the system. The format programs do the actual 'formatting' as well as creating the basic structural information on the disk. The structural data consists of the free sector list, system information record, and the root directory. Disks may optionally have swap space reserved as well.

There are several points to consider when formatting a disk. They are:

1. Will there be a very large number of files on the disk?
2. Will the disk be used on the 'swap device'?
3. Is the disk part of a named set of disks?

The answers to these questions will determine the options to be specified with the format command. Note that some format commands do not accept options. Most of the format programs for hard disks do not accept options, but will automatically provide swap space and a number of files which you are unlikely to exceed.

The floppy disk formatter is an example of one with options. The options allow density, side count, fdn number, and swap space specification. Specifying the number of fdn blocks determines the maximum number of files that may exist on the disk. Fdn stands for 'file descriptor node' and is a system entity used to keep information about a file. There are 8 fdn's per disk block. If you would specify 60 fdn blocks, the disk would be able to support 480 files.

The swap space may be specified on the floppy as well. By default, no swap space is reserved on floppy disks. You should avoid using a floppy as a swap device if at all possible. If you must, use at least a double sided disk, since the minimum amount of swap space should be about 200K. NOTE: Since the system as supplied resides on a single density, single sided floppy diskette, it has limited usefulness because of the small amount of free space and swap space. You should only use this disk as a backup to make new disks on a medium capable of holding more data.

Once a disk has been formatted, it may be 'mounted' on the file system and used. Once mounted, you may create directories and files on it, just as on the original disk. A system program called '/etc/crdisk' has been provided to make a system disk. This program will create all of the standard directories, files, and devices which make up a standard UniFLEX system. Crdisk is a text file consisting of a rather long list of commands. Since the execute permission bit is set on this file, it may be executed by simply typing the file name. As your basic system grows with more files and user directories, you may want to edit the '/etc/crdisk' file and add new commands. Before running 'crdisk', you need to mount a freshly formatted disk on the directory '/usr2'. A typical sequence might look as follows:

```
     ++ /etc/mount /dev/fd1 /usr2
     ++ /etc/crdisk
     ++ /etc/unmount /dev/fd1
```

Note that nothing will be displayed while 'crdisk' is running and that it may take several minutes to run. Studying the '/etc/crdisk' file will probably be quite instructive.

IMPORTANT NOTE! The disks created using the above procedure will only be able to boot the system if they were made from the original system disk purchased. All copied versions of the 'uniflex' file need to be installed, using the 'install' command, before they will function correctly. It is not possible to install a 'copy of a copy' of the 'uniflex' file. In other words, if you need to make a disk from which you will boot the system, you will need to get the actual 'uniflex' file from the original disk purchased, copy it onto the new disk, and then use the 'install' command. The 'install' is the last step performed by 'crdisk'. You should read the description of the 'install' command in 'UniFLEX Utility Commands' for more details. If your original disk ever becomes damaged, you will need to contact Technical Systems Consultants, Inc., for a replacement.

IX. Printer Spoolers

This section is more of a configuration topic but is included in this document since it is easy to do. The printer spoolers on UniFLEX are quite general and easily adapted to various printer drivers. There are two printer drivers provided with the standard system, 'spr' and 'ppr'. They reside in the directory '/dev'. The 'spr' driver is the serial printer driver and 'ppr' is the parallel printer driver. Consult the 'Hardware Setup' sheets for details of your particular standard drivers.

UniFLEX is delivered with a 'core' printer spooler. It must be configured before use. The core spooler is the file named '/etc/print'. To demonstrate spooler configuration, an example will be given. Let's assume we want a serial printer to be driven by a spooler. We will use the device driver '/dev/spr' for this purpose. The first step of the process is to create the spooler command. This command name must be the same as the device driver name, in this case, 'spr'. If this name is not acceptable, you can always rename the driver file to something different. To create the spooler command, type:

```
     ++ link  /etc/print  /bin/spr
```

This command will create a new command file named '/bin/spr' which is simply a link to the spooler core file '/etc/print'. The new file is put in the system '/bin' file since it will be frequently used. The next step of the configuration process is the creation of a spooler directory for this device. Type the following:

```
     ++ chd /usr/spooler
```

You are now in the directory 'spooler' which contains (or will contain) all of the spooler directories. Each spooler needs its own directory. The directory name will be the same as that of the device driver. In this case, type:

```
     ++ crdir spr
```

which will create a spooler directory for the device 'spr', our serial printer driver. That's all there is to it! You now have a new command which is called 'spr' and can be used to spool files to a serial printer. You can also create a spooler for the parallel driver 'ppr' in a similar manner. You can have as many printer spoolers as you have drivers and printers.

Each time the system is booted, it is necessary to invoke the desired spoolers. This is usually done while the system is still in 'single-user' mode. Just having the spooler command and spooler directory will not accomplish anything. The actual spooler program(s) must be invoked before spooling can actually take place. A little background on how the spooler operates may be helpful. Invoking a spooler, as follows:

```
     ++ /etc/insp spr
```

will startup the spooler for the device 'spr'. Spoolers are run in the background and continue running until a 'pstop' command is executed. When 'insp' is run, it will create a file named '.mrk*splr?' in the directory '/usr/spooler/spr', which is the spooler directory for 'spr' that we created above. If the spooler is already running when the 'insp' command is executed, you will get an error message. A spooler is considered running if the 'mrk*splr?' file already exists. The 'pstop' command will delete this file, but under some circumstances, such as system crashes or stopping the system without using 'pstop' to stop the spoolers, the 'mrk*splr?' file may remain in the spooler directory even if the spooler is not actually running. If this happens, it is necessary to 'kill' this file before attempting to invoke the spooler. Note that since the name of this file contains special characters, you must enter the name enclosed in quotes. The spooler comes alive from time to time (about every 15 seconds) and examines its spooler directory. Any files contained in this directory are printed, and deleted. Files are entered into the spooler directory by using the spooler command created above, 'spr'. As an example:

```
     ++ spr file1 file2
```

will print the files 'file1' and 'file2' on the serial printer. Note that the original files will not be deleted after printing, but the new references to them, which were put in the spooler directory '/usr/spooler/spr', will be deleted. Here is another example:

```
     ++ dir +l | spr
```

This command will 'pipe' the output of the 'dir' command into the spooler 'spr'. It is possible to examine the contents of a spooler's print queue (which is just the spooler directory) by doing a 'dir' on that directory.

For more information concerning the operation of the printer spoolers, consult the descriptions of 'insp', 'print', 'purge', and 'print control' in 'UniFLEX Utility Commands'. You may want to restrict the use of the print control commands to the system manager. To do this, simply type the following:

```
     ++ chd /usr/bin
     ++ perms o-rwx end idle next rerun
```

The 'insp' and 'pstop' commands should always be restricted.

X. Overview

The UniFLEX system manager is an important person. Acquiring the required skills will take time. The more you read the system's documentation and actually make use of the system's features, the sharper your skills will become.