Begin {

RE: S-8000 ZEUS ~~UUCP~~ NOTES:

To anyone who has attempted to install UUCP, the following quote from section 3.1 of the **Zeus Utilities Manual** might cause some difference of opinion:

"Installing uucp under ZEUS requires little effort."

"Installing" may be easy...getting it to work is another matter. This note is intended to offer a few tips that may be of help.

First, the function of UUCP is not readily apparent from the available documentation, so let's begin with a little clarification. UUCP is a **communications package** that is capable of performing two functions:

1) transfer files
2) cause the execution of a program on another system

How does it work? At first glance, one might expect UUCP to be interactive in the manner of remote and cu. Not so...UUCP is **batch-oriented**. The typical UUCP sequence of events is as follows:

1) the user invokes the UUCP command by typing something like this:

uucp filename remote_system_name\!/

where filename is the name of the file the user wants to transfer, and remote system name is the name of the system to which the file is to be transferred.

2) the uucp command copies the file into the directory **/usr/spool/uucp** under a special name that UUCP can recognize. Another special control file is created in the same directory.

3) a crontab entry causes **/usr/lib/uucp/uucico** to be invoked at a specified time of the day. uucico looks at the entries in the directory **/usr/spool/uucp** and recognizes the special filenames that UUCP copied into usr/spool/uucp. Using information contained in the control file that is associated with each data file, uucico initiates the call to the remote system and transmits the file.

—>

What about the physical connection between the two machines? The simplest hookup is the direct connect through a **null modem** cable (a cable that switches transmit and receive signals) or a **null modem device.** Anyone who has two machines in close physical proximity should take advantage of "direct connect," because it simplifies debugging.

The physical link can also be established through a modem connection. The actual phone connection must be established through an automatic calling unit (For an exception, please see discussion on debugging).

If you read the documentation, you will learn that UUCP relies on information obtained from several files. In the discussion that follows, the user's system is referenced as the "local" system, and the system with which he intends to communicate, as the "remote" system. The intent is to send a file from the local system to the remote system.

local	remote
/usr/lib/uucp/USERFILE	/usr/lib/uucp/USERFILE
/usr/lib/uucp/L-devices	/usr/lib/uucp/MYNAME
/usr/lib/uucp/L.sys	/etc/passwd
/usr/spool/uucp/SEQF	
/usr/spool/uucp/LOGFILE	
/etc/ttys	/etc/ttys
/dev/ttyX	/dev/ttyX

What about the contents of these files? To minimize confusion, the following contents for each file are suggested. The intent is to get UUCP working; security will be addressed later.

/usr/lib/uucp/L-devices (local)

tty0 0 9600

—→

By referencing tty0, I indicate my intention to use tty0 on the local system to establish communication with the remote system. One end of the null modem cable is connected to the local tty0 and the other end into any any available port on the remote system.

The 0 means that I am not going to use a Bell 801 Automatic Calling Unit (ACU). A zero is appropriate in this position for both direct connect and dialing out through an automatic dialling unit (not to be confused with the ACU).

9600 is the baud rate that I am going to use. This means that tty0 in /etc/ttys should have a 2 in column 2.

/usr/lib/uucp/L.sys (local)

Remote: Any tty0 9600 tty0 login:-x-login:-x-login:-x-login:-x

(cont...) <sp>uucp ssword: uucp

The first entry in the L.sys file is the name of the system that I want to call. In this case I have called the remote system "remote."

Any refers to the time period in which the local system is allowed to call the remote system. "Any" means that a call can be made at any time.

tty0 refers to the device I want to use. An entry for this device must appear in the L.devices file.

9600 is the baud rate I will use. It must agree with the tty0 entry in the L-devices file.

If you are using a direct connect link, the argument following the baud rate is simply a reiteration of the device. It must agree with the first reference to the device.

If you are using a modem connect link, the control sequences and phone number for placing a call through the automatic dialling unit are placed after the baud rate. For example, the Ventel must see two carriage return/linefeeds before it will allow a user to dial a phone number. You must supply that series of control codes and any special commands required by the calling unit at this position in the L.sys file. Doing so is not easy, since the editors apply their own interpretation to the control codes you are trying to enter. However, it can be done using ed in conjunction with a !stty nl from within the editor. An L.sys file for the Ventel 212A Plus is available from the Tech Support Group. Contact Kathy Butler, ext 4000.

--->

For reasons unknown, UUCP (or the Ventel--the culprit has not yet been determined) does not handle phone numbers correctly: it seems to skip every other digit in the phone number. For example, if you wished to dial 97275541, UUCP (again, or the Ventel) would actually dial 9254. To circumvent this problem, supply a dummy character between each digit. For example:

9x7x2x7x5x5x4x1

where x is the dummy character in this case.

login:-x-(etc). You must tell UUCP what to expect when it attempts to establish communication with the remote system. In this case, UUCP will expect to get a "login:", which is just what it should get. The use of "-x-" is not known to be documented, but is believed to simulate the pressing of the return key to ensure that a login message is activated. The manual recommends

login: uucp

but that has not been seen to work.

After the last "login:-x" the "uucp" means that UUCP on my local system will log in on the remote system as "uucp". Therefore, the remote system must contain an entry in the /etc/password file for the user UUCP. The login directory must be /usr/lib/uucp, and, instead of a c-shell, the program /usr/lib/uucp/uucico will be run.

ssword: uucp This tells UUCP to expect a "password" message from the remote system and to send the password "uucp" in response.

Note that UUCP only seems to examine the last few digits of something it expects. Thus, you do not have to supply the fullword, "Password," and you do not have to include the full "ZEUS login:" message in the L.sys file.

/usr/spool/uucp/LOGFILE (local)

When UUCP is activated, it writes rather terse descriptions of its progress in LOGFILE. Make sure that it exists.

/usr/spool/uucp/SEQF (local)

-->

When executing your first UUCP, this file should contain: 0000 UUCP uses this file to create unique file names by incrementing this number each time it prepares a file for transmission.

/usr/lib/uucp/USERFILE/ (local & remote)

This file should contain: , /

This means that any local system can call the remote system and can transfer any file from or to the system. When you have UUCP working, you can specify the name of the login user, the local system that the user is calling from, and the directories this user may access. For example,

uucp,local /tmp

In the above case, the MYNAME file on the local system would contain "local," would specify "uucp" as the login name in the L.sys file on the local system, and would have access to the directory /tmp on the remote system.

/usr/lib/uucp/MYNAME (remote)

This file should contain the name of the system: remote. The contents of this file must match the system name entry in the L.sys file on the local system.

etc/ttys (local & remote)

The /etc/ttys files on both systems are extremely important. In any attempt to establish communication, one of the systems must initiate a call. The system that initiates the call must NOT have a listener forked for the device that UUCP is going to use. The system that receives the call MUST have a listener forked on the device to which the null modem cable is plugged. Therefore the entries for the /etc/ttys files are listed below:

local	remote
02ttyO	12ttyX

where X is the tty that is connected to the local system via a null modem cable or a modem.

} End

Begin {

RE: TROUBLE-SHOOTING ~~UUCP~~

Once you have set up the files, verify that the ttys are running at the desired baud rate and that the remote tty has a listener by plugging a terminal into that port and obtaining a login message. Now you are ready to begin a few test transfers.

First, login as ZEUS on the local system. You will need to examine files to diagnose information, remove some of them, etc., so login with the greatest power available.

Change directory to /usr/spool/uucp and create a file for test-transfer. Then initiate the process by typing:

uucp test remote\!/

where test is the name of the file you wish to send and remote is the name of the system you want to call. The ! is a special character that identifies the system name as a system name, and, when running in the c-shell, it must be escaped in order to avoid the "Event not found" error message. The / is the directory you wish to copy test to, and it corresponds to the entry in USERFILE on the remote system.

After a couple of seconds, your prompt will return. Dog LOGFILE in order to follow the progress of UUCP. If your physical connection is good, you will see a message that the call has been queued followed by a call-succeeded or call-failed message.

While UUCP is active, a lock file is placed on the tty UUCP is using. When the lock file is removed by UUCP, all that is going to happen has happened. If you are extraordinarily lucky, UUCP has transferred your file. On the other hand, you may dog the LOGFILE and find a message such as:

call succeeded. work done (here)

Unfortunately, this only means that it is now time to begin debugging.

Before you begin debugging, you must first clean up after the unsuccessful attempt to transfer the file. The LCK.. tty file may have been left on. Remove it. Second, UUCP leaves a file that starts with STST. In this case, the file would be called STST.remote. Remove it. If you attempt another UUCP before this file is removed, you will get "retry time not reached" in your LOGFILE.

Next, check your device. For an as yet unknown reason, UUCP changes the mode of the tty and leaves it that way when it fails to complete the call. Chmod back to 666.

Check the modes of the device on the remote system, and change them as well, if necessary. Also, do a ps -alx on the remote system. If you should find /usr/lib/uucp/uucico running on your device, kill it. UUCP on the local system cannot communicate with the remote system while a "hung" uucico is incommunicado on the remote port.

The extent of damage depends upon how far UUCP got in establishing communications, so don't be disappointed if all of these things did not occur.

Return to the /usr/spool/uucp directory. An ls -al should reveal two files distinguished by the first two characters of their names:

C. remoten0001
D. remoten0001

The 0001 is a sequence number that UUCP obtained from the SEQF file. It literally means that this is the first file that has been submitted to be transmitted. The next file to be submitted will be given an incremented sequence number, and the number contained in SEQF will be equally incremented.

The file C.remoten0001 contains control information that UUCP uses to establish communications with the remote system. D.remoten0001 is a copy of the file you wanted to transmit. In fact, all the "uucp" command does is create these two files and activate the real worker, /usr/lib/uucp/uucico.

Now you are ready to enter debug mode (which is, unfortunately, so obliquely referenced in the documentation). Type:

/usr/lib/uucp/uucico -r1 -x9&

The r1 causes uucico to be invoked in "master" mode; i.e, a mode in which uucico can initiate a call.

The -x9 is the most verbose debug level. At the present time, the only other known debug level is 4, which provides information on the login process only.

This invocation of uucico will cause uucico to try to send the file again and in the process will yield information that is invaluable in debugging. You will see what uucico is actually sending and receiving during the dial and login process.

If your automatic dialling unit is preventing you from debugging UUCP, you can try a trick developed by your Zilog SAE in Dallas: use remote to log onto the system with which you want to use UUCP. Then enter a local command (without the -l option) to exit from the remote system while retaining the open line. Then use /usr/lib/uucp/uucico to attempt file transfers. To use this technique, you must provide the device name instead of phone number in the L.sys file. —>

We should like to be able to provide more information on what to look for in debugging, but the subject is simply too vast to be attempted in a note like this. Remember to keep your initial attempts at UUCP simple. Remember that the contents of the USERFILE on both systems are absolutely critical to success. Access to all directories in the pathname is an often over-looked failure point. Invisible and unwanted control-characters in the L.sys, L-devices, and USERFILE files have been known to make UUCP choke.

When all else fails, give us a call. Good luck!

} End

Begin {

RE: A NOTE ON THE ZEUS COMMUNICATIONS PACKAGE ~~CU~~

Many users have been having trouble getting CU to work properly. If you carefully follow the ZEUS REFERENCE MANUAL, section 1, documentation, and enter your command line as

cu -s 1200 -l tty2 dir

you will be given the message:

tty2 not known at 1200 baud

This is because the 2.1 version of CU checks the file /usr/lib/uucp/L-devices to verify the parameters on your command line. The entry in the L-devices file for tty2 should look like this:

DIR tty2 1200

Such an entry will cause CU to work correctly with the above command line.

Unfortunately, another communication package, uucp, also looks at the L-devices file. Of course, uucp expects different parameters, and will halt with a

not available (device)

message in the LOGFILE if uucp cannot find the entry it is looking for. The temporary solution is to have two entries for each device capable of dialling out.

DIR tty2 1200	(looks good to cu)
tty2 0 1200	(looks good to uucp)

This is a temporary work-around. We hope to have these two programs agree in future releases.

} End