

Escuela Superior Politécnica del Litoral

FACULTAD DE INGENIERIA EN ELECTRICIDAD

INTERCONEXION DE DOS SISTEMAS UNIX POR MEDIO DE UUCP

Realizado por:

CARLOS J. VELAZCO PLAZA

ESTUARDO SEVILLA JARA

WISMARK A. RAMIREZ GORDILLO

Director del Proyecto:

ING. JAIME PUENTE PEREZ

**Informe Final previo a la Obtención del Título
de Ingeniero en Electricidad
Especialización Electrónica**

1993

DEDICATORIA

Damos gracias a Dios por habernos permitido culminar nuestra carrera de Ingenieros en Electricidad.

Este trabajo esta dedicado a nuestros familiares, en especial a nuestros queridos padres, ya que con su apoyo hemos podido lograr este objetivo.

RECONOCIMIENTO

Dejamos constancia de nuestro agradecimiento a los profesores y a todas las demás personas que han colaborado para nuestra formación profesional.

Agradecemos especialmente al Ing. Jaime Puente Pérez por toda la colaboración, ayuda y consejos dados para la culminación de este proyecto.

INDICE

CONTENIDO	PAG.
INTRODUCCION	1
Contenido de este informe	2
Versiones del UUCP	3
CAPITULO 1: UNIX TO UNIX COPY (UUCP)	5
1.1 Características generales	5
1.2 Hablando sobre el demonio uucico	6
1.3 Archivos de configuración	7
1.4 Como trabaja una conexión	9
1.5 Configurando un enlace	12
1.6 Archivos y directorios	14
CAPITULO 2: HABILITANDO EL UUCP	16
2.1 Instalación	16
2.2 Hardware	17
2.3 Software	19
2.3.1 Configurando un puerto en la Sun	21
2.3.2 Configurando un puerto en la U6000	23
2.4 Configurando los archivos de instalación UUCP	24
2.4.1 Editando el archivo Systems	25
2.4.2 Editando el archivo Devices	27
2.4.3 Editando el archivo Permissions	29

2.5 Editando el archivo /etc/passwd	31
2.6 Resumen	33
CAPITULO 3: MANUAL DEL USUARIO	35
3.1 Trabajando con el comando cu	36
3.1.1 Conectándose con el comando cu	36
3.1.2 Secuencias de escape tilde (~)	37
3.1.3 Transfiriendo archivos con cu	38
3.2 Trabajando con el comando mail	38
3.3 Trabajando con los comandos de transferencia de archivos	39
3.3.1 Los comandos uuto y uupick	40
3.3.2 El comando uucp	41
3.4 El comando Uutry	43
3.5 Procedimientos recomendados	43
CONCLUSIONES Y RECOMENDACIONES	45
BIBLIOGRAFIA	47
ANEXOS:	48
ANEXO A: Referencia rápida de comandos UUCP	49
ANEXO B: Manual de fallas	54
ANEXO C: Glosario	58

INTRODUCCION

La ESPOL se encuentra conectada a la red Internet por medio de una computadora Sun la cual se encuentra ubicada en Cesercomp. La idea sobre este proyecto es de que haya acceso a la red Internet desde el laboratorio de microcomputadores, para lo cual se habilitará al minicomputador **Unisys U6000** de manera que emplee el paquete de programas UUCP, esto es *Unix to Unix Copy*, para comunicarse con el equipo **Sun**.

UUCP es un paquete de programas que viene instalado como una utilidad más del sistema operativo Unix, proporcionándonos las siguientes facilidades:

- Transferencia de archivos entre sistemas Unix (**uucp**).
- Ejecución de comandos remotos (**uux**).
- Envío y recepción de correo a sistemas remotos (vía **mail**).

La gran ventaja del **UUCP** sobre muchas otras tecnologías es de que no se requiere de fuertes inversiones de hardware y software para realizar estos cometidos, ya que está diseñada para usar cables seriales estándares, modems y el servicio telefónico común.

Cada sistema en una red UUCP tiene archivos que describen a los otros sistemas directamente conectados a él, y qué tipo de conexión es posible con ellos. La implementación de esos archivos es una de las mayores tareas en la instalación de una red UUCP. Una vez que esto esté establecido, el UUCP requerirá de mínima supervisión.

Contenido de este informe.-

El presente texto se encuentra dividido en capítulos en los cuales podremos encontrar:

- Capítulo uno:** En este capítulo se encontrará una descripción de como trabaja el UUCP, dando el concepto de la manera en que se transfieren archivos, haciendo énfasis sobre como los programas UUCP hacen uso de los diferentes archivos de configuración, que se necesitan editar.
- Capítulo dos:** En este capítulo tratamos todo lo referente a la instalación del UUCP tanto en hardware como en software, detallando los formatos de `clu` de los archivos que fué necesario editar y lo que se hizo en ellos.
- Capítulo tres:** En este capítulo se encuentra el manual del usuario, con descripciones y ejemplos de cada comando, así como una guía de los usos que en este proyecto se le pueden dar.
- Anexo A:** En este anexo se encuentra una guía rápida de los comandos existentes en UUCP, así como de las distintas opciones de `clu` de ellos.
- Anexo B:** En este anexo se encuentran una serie de mensajes de errores de la conexión, los cuales dan un indicativo sobre porqué una transferencia no se puede realizar, sean estos físicos o de hardware. Además en este anexo se encuentra una guía de solución de fallas.
- Anexo C:** En este anexo se encuentra un breve glosario con los términos cuyo significado hemos considerado conveniente aclarar.

Versiones del UUCP - En esta investigación nos dimos cuenta de que existen muy variadas versiones de UUCP, cada una con sus ventajas y desventajas, y cada una con sus diferencias respecto a la otra, por lo que resulta de crucial importancia saber qué versiones son las que se encuentran en el hardware que se emplee, para poder proceder correctamente.

La primera versión de un sistema UUCP fue elaborada en 1976 por Mike Lesk en los laboratorios AT&T Bell. Fue tal su éxito que posteriormente se desarrolló una versión mejorada, la que se conoce como UUCP Versión 2. Una serie de actualizaciones fueron incorporadas por los laboratorios AT&T en las versiones de UNIX System Y Release 1 (SYR1) y System Y Release 2 (SYR2).

Mientras tanto se crearon unas actualizaciones por investigadores independientes en las Universidades de Duke y Berkeley, las cuales fueron la base de las versiones conocidas como Berkeley Software Distribution (BSD 4x), y también de las versiones para equipos Ultrix (DEC) y SunOS (SUN).

Con la versión UNIX System Y Release 3, AT&T comenzó a distribuir una nueva versión de UUCP conocida como **BNU** ("Basic Networking Utilities"). Esta versión fue desarrollada en 1983 por Peter **Honeyman**, David A. **Nowitz** y Brian E. **Redman**, quienes reescribieron el UUCP de la versión 2, quitándole ciertas deficiencias, proporcionándole un menú de administración que facilita mucho las cosas y proveeyéndole de soporte para nuevas y más avanzadas redes y dispositivos de comunicación. Esta versión también se conoce como UUCP **HoneyDanBer** (viene de los nombres de sus creadores). La versión BSD 4.3 posee unas actualizaciones muy significativas, incorporando algunas de las mejoras de BNU pero manteniendo una mayor continuidad con las otras implementaciones de la Versión 2. Algunos equipos SunOS se han cambiado a esta versión (BSD 4.3).

Para complicar más las cosas existen una serie de pequeñas diferencias en la implementación de las distintas versiones por parte de los fabricantes de equipos. Por

ejemplo, Xenix usa la versión SYR2 UUCP, pero añadiéndole algunos detalles de BNU.

En la investigación desarrollada se descubrió que la **U6000** posee un UUCP versión **BNU**, mientras que la **SunOS**, una versión UUCP **BSD 4.3**.

CAPITULO 1

UNIX TO UNIX COPY (UUCP)

Este capítulo trata sobre como trabaja el UUCP en términos generales y explica los mecanismos que se utilizan durante una conexión UUCP y que son transparentes para el usuario de este paquete de programas. Además nos referiremos brevemente a los archivos empleados para instalar la conexión UUCP.

1.1 Características Generales

Como ya hemos enunciado, UUCP significa *Unix to Unix CoPy*. Este paquete consta de un grupo de programas, siendo los tres principales:

- **uucp**. El programa uucp trabaja de una manera muy similar al comando cp (copy) de Unix, proporcionando transferencia de archivos entre sistemas UNIX. El nombre de este programa, por ser el principal, se ha tomado para bautizar al paquete.
- **uux**, el cual es un programa usado para hacer pedidos de ejecuciones de comandos en sistemas remotos. Por ejemplo, un pedido de impresión de un archivo en una impresora conectada a una máquina remota.
- **mail**, el cual no es normalmente pensado como un programa UUCP, sin embargo comprende la sintaxis de UUCP para direccionarse a sistemas remotos, y está muy relacionado con UUCP sobre todo sistema.

Las diferentes versiones de UUCP pueden soportar muchos programas adicionales; pero por lo menos los tres programas enunciados se encuentran en todos los UUCP.

Un programa que corre en background (demonio), llamado **uucico**, es el que en verdad ejecuta la mayor parte del trabajo de transferencia de archivos o de pedido de ejecuciones remotas entre sistemas. Otro demonio, el **uuxq**, es invocado en el sistema remoto para procesar los requerimientos de ejecuciones remotas. En adición a estos demonios, se tienen varios archivos de datos con información sobre los sistemas que pueden ser llamados, y los mecanismos para hacerlo.

1.2 Hablando sobre el demonio uucico.-

El UUCP es una red de "almacenar-y-continuar" (store-and-forward). Es decir, al hacer un pedido de transferencia de archivos o de ejecución remota de comandos en otro sistema, esto no se realiza inmediatamente, sino que el pedido que sea, es almacenado en una memoria temporal (es decir, es "spooled"), para su posterior realización cuando la comunicación sea establecida entre los dos sistemas.

El programa **uucp** propiamente dicho no copia los archivos de un sistema a otro, ni el **uux** en realidad ejecuta comandos en un sistema remoto, sino, que cuando un usuario invoca a cualquiera de estos dos programas, o envía correo a otro sistema, ocurren dos cosas:

1.- Se crea un archivo de trabajo (*uuc*), el cual contendrá información como los nombres de los archivos fuente y destino, las distintas opciones que dan los programas **uucp** y **uux**, y el tipo de pedido (enviar, recibir o ejecutar). Este archivo de trabajo se crea dentro del directorio **/usr/spool/uucp**. Dependiendo de nuestra versión de UUCP y de las opciones de comando que estemos empleando se creará

también un archivo de datos (data file), el cual contendrá una copia del archivo a ser transferido. Los archivos de datos también se crean siempre que se mande correo a alguien en una localidad remota o se haga un pedido de ejecución de un comando remoto.

2- El programa demonio **uucico** es invocado y es el que en realidad realiza la transferencia.

Cuando el demonio **uucico** es invocado, busca en el directorio spool por algún archivo de trabajo y por intentos de contactar otros sistemas y realizar las ejecuciones de instrucciones en archivos de trabajo (en nuestra versión BNU, un programa llamado **uusched** es el que realiza esta búsqueda y llama al **uucico** solo cuando se necesita llamar a otro sistema).

Sin embargo, los archivos de trabajo solo contienen una pequeña cantidad de la información que el **uucico** necesita para realizar la transferencia. Ellos le dicen al **uucico** qué hacer, pero no cuando y como hacerlo. Esta información está contenida en la Unisys en un conjunto de archivos de configuración (configuration files) dentro del directorio **usr/lib/uucp**, y en la Sun ese conjunto de archivos de configuración están dentro del directorio **etc/uucp**. Estos archivos son los que como administradores de la red UUCP deberemos configurar.

1.3 Archivos de configuración

El más importante de estos archivos es el llamado **Systems**, tanto en la Sun como en la Unisys.

Este archivo contiene una lista de los sistemas conocidos por el sistema local, es decir una lista de los únicos sistemas con los que es aceptable y permitida la comunicación, y además las instrucciones para poder llegar a ellos; por ejemplo:

```
espol Any sun 19200 - ogin: -- ogin: nuucp ssword: uucpuucp
```

Esta línea nos dice que el sistema espol:

- Puede ser llamado a cualquier hora del día (**Any**).
- Usando un dispositivo al que se ha llamado **sun** (podría ser otro nombre).
- Con una velocidad de transmisión de **19200** baudios.
- El **"guión"** nos indica que no es necesario marcar ningún número telefónico (conexión directa).
- conectándonos con el login name **nuucp**.
- y el password **uucpuucp**.

En el caso de haber múltiples maneras de llegar a otro sistema (ej: Dos números telefónicos), habrán múltiples entradas para ese sistema en **Systems**.

Otro archivo de configuración muy importante es **Devices**, el cual contiene un mapeo uno-a-uno entre el hardware de comunicaciones usado por nuestro sistema (tal como un modem, una conexión de red o una línea directa serial) y el nombre del dispositivo que el **uucico** deberá usar para acceder a ese dispositivo. Por ejemplo, en este archivo encontramos una línea:

```
sun tty100 - 19200 direct
```

Esto nos informa que el **uucico** puede usar el nombre **direct** para referirse a una conexión directa en el puerto serial **tty100**.

Si la conexión entre sistemas es vía modem, el **uucico** deberá ver la manera de como llamar a ese modem. Las instrucciones para llamado a los modems se encuentran o codificadas en el UUCP o contenidas en un archivo de accesorios (en el BNU se hallan en el archivo **Dialers**). Si no se encuentra configurado en los archivos de hardware el modem que vamos a emplear, se deberán añadir instrucciones de llamado para ese tipo de modem.

Además existen archivos que controlan qué parte de nuestro sistema de archivos será accesible a cada sistema remoto. Esto es por propósitos de seguridad. En la versión BNU este archivo se llama **Permissions**. Hablaremos con mayor detalle sobre estos archivos de configuración más adelante.

1.4 Como trabaja una conexión

El **uucico** usa la parte final de la línea del archivo **Systems** para conectarse con un sistema remoto. Esta parte tiene campos "espera" y "envía", por ejemplo:

- esperado "....ogin:"
- cuando se recibe "ogin:", se envía "nuucp"
- esperado "....sword:"
- cuando se recibe "sword:", se envía "uucpuucp"

Así, se efectuará una conexión. Por supuesto que esto presupone que en el sistema remoto hay una identificación de login en el archivo `/etc/passwd` para un usuario llamado **nuucp** cuyo password es **uucpuucp**, y, como una acotación de gran importancia, el shell de login de este usuario no debe ser otro que el **uucico**. Este tipo de shell es un entorno especial, diferente del C shell o el Bourne shell, y es empleado para realizar una conexión automática.

Clave:

Los dos programas **uucico**, uno en cada sistema que se están comunicando, trabajan en tandems para realizar la transferencia. Primero se presentan entre ellos; ambos sistemas deben de reconocerse, esto es, espol debe tener una entrada para **uucico** en su archivo **Systems** y viceversa.

Luego, los dos sistemas deben elegir qué protocolo usarán. La elección de este generalmente depende del tipo de enlace físico que se usa en la comunicación. En nuestro caso, no se utilizó ningún protocolo por tener una línea dedicada y confiable.

Clave:

Después de esto, la transferencia ocurrirá. El **uucico** del sistema que llama está en su rol "master", (es decir, controla el enlace, especificando qué se desea, esto es, enviar datos, recibir datos o ejecutar comandos remotamente), mientras que el del sistema que recibe la llamada está en el rol "slave", solo verificando permisos para ver si el master tiene "derecho" a lo que pide o no. Si es así, la transferencia comienza.

El sistema que envía lo hace por paquetes de datos, el que recibe envía reconocimientos por cada paquete recibido. Mientras este archivo es recibido, se almacena en un archivo temporal del directorio **spool** (buffers) del sistema que lo recibe. Una vez finalizada la transferencia, el **uucico** del sistema que recibió lo copia del archivo temporal al destino requerido.

Clave:

Si alguno de los requerimientos procesados fuera de una ejecución de comandos remotos (**uux**), un archivo ejecutable se creará en el directorio **spool** del sistema receptor.

Clave:

Cuando el **uucico** del sistema que llama finaliza con sus requerimientos, envía una señal de requerimiento de colgado (hang-up) de la llamada. Si el **uucico** del sistema receptor por ejemplo necesita un archivo del sistema que llama para poder realizar una ejecución de comandos remota, se niega a colgar y se invierten los roles master-slave.

Cuando todas las transferencias finalizan, ambos **uucico's** accederán a colgar. En este instante un demonio llamado **uuxqt** comienza a trabajar buscando en el directorio spool por algún requerimiento de ejecución remota. **Uuxqt** leerá los archivos ejecutables en el directorio spool, verificará que el sistema remoto que hizo el requerimiento tenga permiso para hacerlo, y realizará requerimientos de transferencia de archivos si lo necesitase para la ejecución remota. Si todo va bien **uuxqt** hace que un comando realice lo que se requirió remotamente.

Veamos un ejemplo en el que un usuario en el sistema local **Unisys** quiere realizar la transferencia de un archivo llamado **prueba** hacia un sistema remoto llamado **espol**, empleando el comando **uucp**.

1.- un comando es realizado por el usuario:

```
$ uucp prueba espol!usr/spool/uucppublic
```

2.- Se crea un archivo de trabajo (work file) en el sistema local (Unisys). Se invoca al demonio **uucico** local, el cual busca en el directorio spool local y halla el archivo de trabajo.

```
$ c.espolA143759
```

3.- El **uucico** lee el archivo **Systems** para ver si la máquina local "conoce" a la máquina **espol**.

```
espol Any sunout 19200 - ogin: nuucp ssword: uucpuucp
```

fig # 1.1 Ejemplo de un archivo **Systems**

4.- Después de comprobar en el archivo **Systems** que sí se "conoce" a **espol**, y de que está permitido llamar en ese momento, el **uucico** lee en el archivo **Devices**, buscando si existe un dispositivo sunout que emplee esa velocidad de 19200 baudios, y de existir, saber en que puerto está colocado, y que tipo de conexión se emplea.

```
sunout tty100_M direct 19200 ID
```

fig # 1.2 Ejemplo de un archivo **Devices**

5.- De usarse modems para la conexión, **uucico** consultará la información compilada en el archivo **Dialers**. Pero como en este caso la conexión es directa, esto no se realiza, sino que el **uucico** local se conecta con el sistema remoto (**espol** en el ej.), empleando las instrucciones finales del archivo **Systems** (estas se conocen como el "chat script", o sea "escrito de la conversación").

```
...ogin: nuucp ssword: uucpuucp
```

6.- En el sistema remoto (en este caso, **espol**), su **uucico** verificará si nuestro sistema local (en este ejemplo, Unisys) tiene permisos para conectarse, y de ser así, la transferencia comenzará.

```
"Unisys tiene permisos para enviar archivos al directorio uucppublic"
```

fig # 1.3 Ejemplo de comunicación entre **uucico's**

1.5 Configurando un enlace.-

Con estos conocimientos básicos sobre lo que ocurre en una transferencia, vemos que para inicializar un enlace como el de nuestro proyecto, necesitaremos:

1- Establecer un enlace de comunicación físico entre los dos sistemas en cuestión, en nuestro caso usando TAU's y línea telefónica. Una explicación sobre qué es un TAU se encuentra en el anexo C.

4.1.1. Nombres

2- Dar a nuestro sistema un nombre que lo identificará sobre la red UUCP. En nuestro caso los nombres que se emplearon fueron los ya existentes en cada sistema, siendo para la Sun, el nombre **espol**. Para la U6000 se usó el nombre **Unisys**.

4.1.2. Entradas

3- Crear las entradas en los archivos **Systems** que describirán cuando y como poder alcanzar a los otros sistemas con los que nos queremos comunicar. Los otros sistemas deberán hacer lo propio.

4.1.3. Dispositivos

4- Crear entradas en el archivo **Devices** para hacer que el UUCP sepa si usamos una línea de conexión directa o un modem, que puerto serial usamos, y, si usamos modem, que tipo de modem es (ya que diferentes modems requerirán diferentes comandos).

4.1.4. Modem

5- Si emplearemos un modem que nuestro sistema no conoce, deberemos escribir instrucciones de discado para ese modem.

6- Poner algunos mecanismos de seguridad, como cuales archivos de nuestro sistema son accesibles para sistemas remotos, qué comandos pueden ejecutarse remotamente y crear logins id en **/etc/passwd**.

Esto como generalidad, los procedimientos exactos difieren de acuerdo a la versión de UUCP que tenga cada sistema.

4.2. Configuración

4.2.1. Tablas

1.6 Archivos y directorios.-

UUCP tiene 3 directorios principales:

- **/usr/lib/uucp**, el cual contiene los demonios UUCP, los shell scripts administrativos, y los archivos de bases de datos tales como **Systems** y **Devices**. Para la SunOS, este directorio solo contiene los demonios UUCP. Los shell scripts administrativos y los archivos de bases de datos se encuentran en el directorio **/etc/uucp**.
- **/usr/spool/uucp**, el cual es usado para almacenar los archivos de trabajo, archivos de datos y archivos de ejecución para las transferencias "spooled". Este directorio es conocido comunmente como el directorio **spool**. Para la Sun este directorio es el **/etc/spool/uucp**.
- **/usr/spool/uucppublic**, el cual es conocido como el directorio público, o sea el **PUBDIR**. Este directorio es usado para asegurar que por lo menos haya un lugar en cada sistema en el cual UUCP puede transferir y copiar archivos.

Es de anotar que dentro del directorio **/usr/spool/uucp** se encuentran los archivos de status y administración uucp, como se ve en la siguiente tabla:

<u>Archivo</u>	<u>Versión BNU y BSD 4.3</u>
<i>Error log</i>	.Admin/errors
<i>Log file</i>	.Log/uucico/system
	.Log/uucpl/system
	.Log/uux/system
	.Log/uuxqt/system
<i>Status file</i>	.Status/system
<i>Transfer Statistics</i>	.Admin/xferstats

Tabla 1.1 Archivos de status y administración UUCP

Los archivos *Error log* y *Logfile* son considerados **archivos LOG**, los cuales contienen un record de las llamadas a sistemas remotos, requerimientos encolados, ejecuciones del comando **uux**, y los resultados de transferencias de archivos.

El archivo *Status* contiene información acerca del status de conversación entre dos sistemas.

El archivo *Transfer Statistics* contiene estadísticas de conexión y tráfico para usar con **uustat**.

Además se crean **LOCK Files** por parte del **uucico** en el directorio **usr/spool/locks**, cuando se utiliza un dispositivo. Estos archivos impiden que se realicen múltiples intentos de usar el mismo dispositivo. El **uucico** borra esos archivos cuando un proceso se realiza normalmente. Cuando un proceso o conversación finaliza abruptamente, estos archivos pueden permanecer en el directorio spool e impedirán realizar futuras transferencias, por lo que deben ser removidos.

CAPITULO 2

HABILITANDO EL UUCP

Para habilitar el UUCP se requiere un mínimo de infraestructura física (hardware), en forma general, un cable RS-232 o un par de modems y una línea telefónica.

En la parte de software, hay que escribir en algunos archivos que son propios del sistema operativo de las máquinas Unix. La forma de escribir en estos archivos, además de los procesos que se realizan durante una conexión UUCP, están detallados a continuación.

2.1 Instalación.-

El UUCP está diseñado para básicamente tres tipos de conexiones :

- Directa (vía cable RS232).
- Por medio de modems (vía línea telefónica).
- Red de área local

Para cada tipo de conexión se debe tener en cuenta parámetros físicos (hardware) y de software.

En la interconexión de nuestros dos sistemas no se empleó ninguno de esos tipos de conexión, sino que se utilizó la infraestructura existente en la Politécnica del Sistema Integrado de Voz y Datos, a través de colocarnos una línea telefónica dedicada entre el laboratorio de Microcomputadores y CESERCOMP. En cada

extremo de la línea se conectaron TAU's Ericsson , y por medio de las facilidades que presta el PBX se simuló una conexión null-modem, es decir una conexión cruzada via RS-232, la cual es la idónea para conectar dos DTE (Data Transmission Equipment) entre sí.

Como se ve, nuestro caso es una mezcla de los dos primeros tipos de conexión, ya que aunque el software está referido para una conexión directa, físicamente se está utilizando una línea telefónica dedicada, aunque no se estén usando exactamente modems.

Es decir que si la teoría está correcta, nuestros dos sistemas en la práctica "parecerán" que se encuentran conectados con una conexión directa, ya que la conexión por medio de Tau's está implementada de manera que sea "transparente" para ellos.

Para mayor claridad dividiremos nuestro informe sobre la instalación de la conexión UUCP en dos grandes partes:

- Hardware
- Software

2.2 Hardware.-

Aunque una conexión directa en UUCP se entiende como un cable RS-232, nosotros utilizamos como ya se dijo anteriormente, dos TAU's Ericsson (uno para cada servidor), conectados a través de una línea telefónica dedicada de cuya conexión y establecimiento de parámetros de comunicación (velocidad, número de bits, etc.) se encargó el PBX.

Para realizar la conexión lo primero que hicimos fue construir los cables adecuados para interconectar los servidores con los TAU's. Teniendo en mente que los TAU's tienen puertos DB-25 hembras, y el puerto seleccionado en la U600 es un

DB-9 macho, se construyó con: Un cable multipar, un conector DB-9 hembra y un DB-25 macho, un enlace directo siguiendo la siguiente tabla de conversión que permite mantener el estándar RS-232:

<u>DTE de 9 pines</u>	<u>Función</u>	<u>DTE de 25 pines</u>
1	DCD	8
2	RD	3
3	TD	2
4	DTR	20
5	GND	7
6	DSR	6
7	RTS	4
8	CTS	5
9	RI	22

Tabla 2.1 Conversión del estándar RS-232 de 25 pines a 9 pines

Para construir el cable de la Sun, simplemente empleamos un cable plano de 25 líneas y dos conectores DB-25 machos tipo cuchilla, quedando el cable configurado punto a punto.

Una vez que tuvimos hechos los cables para la conexión, debimos encontrar un puerto serial que se halle disponible, o hacer a uno disponible, para cada servidor, quizá removiendo un terminal.

El diagrama de la conexión realizada quedó de la siguiente manera:

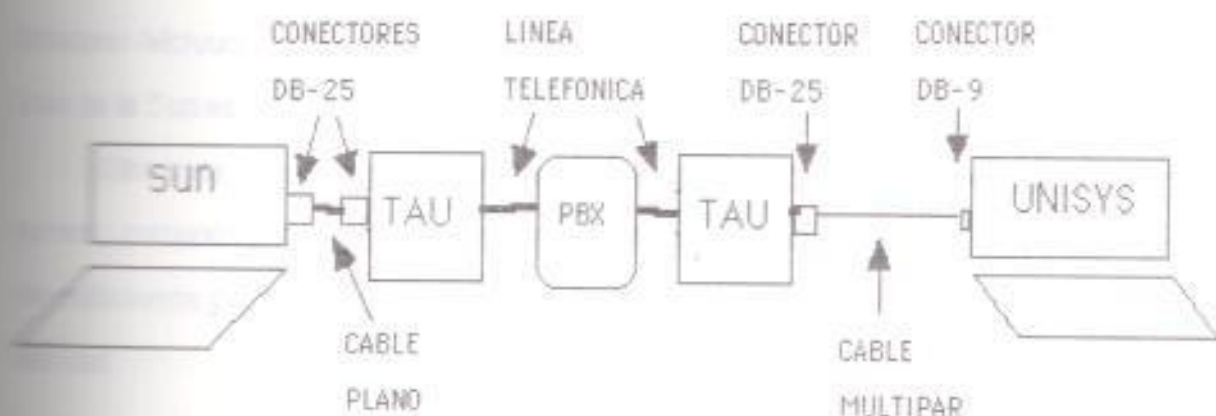


fig 2.1 Diagrama de la conexión

2.3 Software.

Hay que tener presente que para instalar el UUCP, se debe configurar el software en ambas máquinas; de no hacerlo, no se podrá realizar la conexión. Esto también implica que cada sistema puede decidir con quién conectarse y con quien no.

El primer paso consistirá en averiguar con qué versión del UUCP contamos. La forma de conocer qué versión de UUCP es la que se posee, si no viene esta información en los manuales, es:

Utilizando el login name y password del administrador de la Unisys entramos al root, y una vez en la raíz nos cambiamos al directorio `/usr/lib/uucp` con el comando `cd`. Esto lo hicimos para ver si había cierto archivo llamado `L.sys`. (De haber estado, quiere decir que la versión del UUCP es la 2, esto es, la versión vieja). En vez de ese archivo encontramos el archivo `Systems`, lo que nos indica que en la Unisys

tenemos la versión BNU, la cual comenzó a ser distribuida por la AT&T con la versión del Unix System V Release 3.

En cuanto a la Sun averiguamos que el archivo `Systems` está dentro del directorio `/etc/uucp`, e investigando en el `Administrators Guide` averiguamos que el Unix de la Sun es la versión BSD 4.3.

Otra cosa que descubrimos después de algunos intentos con nuestros login names comunes en la Unysys, es de que tanto para ésta como para la Sun, necesitaríamos privilegios de superusuarios para poder realizar las configuraciones debidas.

Una vez realizada la conexión física, hay que tener bien en cuenta el nombre referido en software del puerto serial físico utilizado.

En los sistemas UNIX, los puertos seriales y todos los otros dispositivos, se encuentran dentro de archivos especiales en el directorio `/dev`. Las líneas seriales usualmente tienen nombres que comienzan con las letras `tty`. Para nuestro caso, en la Sun empleamos el archivo `/dev/ttya`, pues al puerto `ttya` estaba conectado físicamente el TAU. En la Unisys U6000 en cambio utilizamos el archivo `/dev/tty100`.

Lo primero que hay que hacer con estos archivos es cambiarle el propietario (owner) de manera que sean UUCP, para lo cual utilizamos el siguiente procedimiento en la Sun (se realiza lo mismo para la U6000):

1- Liste el archivo `tty` a utilizar, para estar seguro de que el puerto existe dentro del directorio `dev`. Esto se hace usando el comando `ls` (list):

```
digite: ls -l /dev/ttya
```

respuesta de la Sun:

```
crw-rw-rw- 1 root wheel 0,8 Jun 26 17:11 /dev/ttya
```


2- Utilice el comando **chown** (change owner-cambio de propietario) con el siguiente formato:

digite:

```
#chown uucp ttya
```

Para la comunicación UUCP se necesita siempre un proceso de conexión, el cual puede ser de dos tipos: automático, empleando un demonio llamado **uucico**, o manual, utilizando el comando **cu**. Es por esta razón que se necesita utilizar un programa llamado **getty**, que se encarga de monitorear los puertos de los sistemas UNIX, y en caso de sentir una señal DCD, colocará un login prompt, para iniciar el proceso de conexión.

Teniendo en cuenta que los sistemas operativos UNIX de la Sun y la Unisys son diferentes, hemos creído conveniente subdividir la sección de como definir el puerto a usar para cada máquina:

2.3.1 Configurando un puerto en la Sun.-

Para habilitar un getty en cualquier puerto de la Sun, se debe de escribir una línea en el archivo **ttytab** que se encuentra en el directorio **/etc**.

Sobre como y qué escribir en dichos archivos hablaremos más adelante.

El **getty** solo se requiere para llamadas que entran en nuestro sistema, mas no para llamadas que hagamos de nuestro sistema local a otro remoto. Para el caso de llamadas que salen no debe existir un **getty** corriendo en nuestro puerto, pues al detectar nuestra propia señal DCD y si un remoto se está intentando conectar con nosotros, se provocaría que nuestro sistema se "quede colgado", ya que los dos sistemas mandarían "login prompts" al sistema local.

Para evitar esta situación, se debe deshabilitar el **getty** cuando se realiza una llamada. Hasta el sistema Y versión 2.1, se debían usar dos puertos para hacer una conexión bidireccional: Uno para llamadas que salen y otro para llamadas que entran.

Afortunadamente, las nuevas versiones de Unix poseen mecanismos para realizar conexiones bidireccionales en un solo puerto.

Así la versión BSD 4.3 de la Sun proporciona el siguiente mecanismo: El driver *serial* soporta dos "puntos de entrada" para cada puerto, para lo cual (*nota: esto solamente puede hacerlo un superusuario):

1.- Creamos un driver *ttyd0*, el cual será empleado para llamadas que entran, simplemente "linkeandolo" (añadiéndole uno o más nombres a un archivo existente) al puerto *ttya* en el directorio *dev*, de la siguiente manera:

```
ln -s /dev/ttya /dev/ttyd0
```

2.- Creamos un driver *cua0* con el comando *mknod*, que físicamente es el mismo que el *ttyd0*, pero que podrá ser usado para llamadas que salgan, realizando lo siguiente:

```
#mknod /dev/cua0 c 12 128
```

3.- Habilitamos el driver *ttyd0* para llamadas que entran, colocándole un *getty* que comience el proceso de conexión, para lo cual se añadió una línea en el archivo */etc/ttytab* de la Sun:



fig 2.2 Edición del archivo */etc/ttytab* de la Sun

devname El nombre del archivo especial en el directorio **dev** que corresponde al dispositivo.

<i>comando</i>	El comando que es ejecutado cuando el DCD es detectado en la línea. Por ejemplo, sobre esta línea corre un proceso <code>getty</code> y la velocidad de la línea es de 19200 baudios.
<i>tipo de terminal</i>	El nombre del terminal conectado a la línea.
<i>status</i>	Se coloca on u off indicando el status del proceso.

4.- Deshabilitamos para la versión driver `ldev/cua0` lo que Sun llama el "software carrier detect", el cual es un programa que, como su nombre indica detecta si hay una señal DCD en el puerto en que corre. Al hacer esto, el `getty` de la línea no detecta si hay DCD y no iniciará ningún proceso de conexión, permitiendo así que llamadas salgan por este driver:

```
#ttypsoftcar n ldev/cua0
```

5.- Una vez realizados todos estos procedimientos, se digitará:

```
#kill -hup 1
```

Para forzar al `init` a releer el archivo `/etc/ttytys`.

2.3.2 Configurando un puerto en la U6000.-

Para habilitar un `getty` en cualquier puerto de la U6000 se debe escribir una línea en el archivo `inittabs` dentro del directorio `/etc`.

El formato del archivo `inittabs` es parecido al `ttys` de la Sun, pero difiere ligeramente como vemos en la línea que se editó en la Unisys.

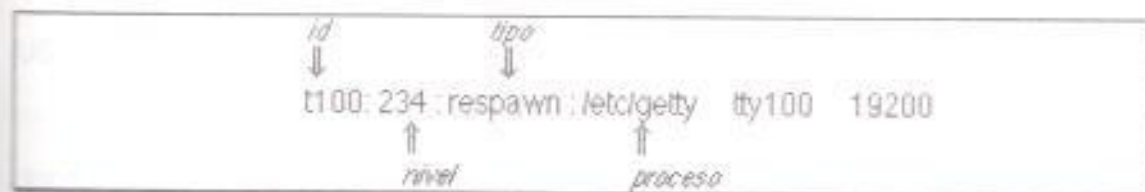


fig 2.3 Edición del archivo `/etc/inittab` de la U6000

id Un identificador de 1 a 4 letras usado por `init` para identificar el puerto.

nivel Una cadena de caracteres (del 0 al 5, o de `a` a `c`), que dan el nivel de ejecución del proceso.

tipo Es como `init` tratará al proceso. Puede ser: `off`, `once`, `wait`, `respawn`, `boot`, etc.

proceso El programa invocado cuando `init` activa la entrada. Para líneas seriales, este es **getty**, el cual pondrá el login prompt.

Después de editar este archivo, se emite el comando `init q` para forzar al sistema a releerlo y hacer los cambios.

2.4 Configurando los archivos de instalación UUCP.-

Como se dijo anteriormente, existen tres archivos principales sobre los cuales nosotros trabajamos. Estos son los mismos para nuestros dos sistemas a conectarse, y son:

- **Systems**
- **Devices**
- **Permissions**

Estos archivos se encuentran en el directorio `/etc/uucp`, para la Sun. En la U6000 se hallan en el directorio `/usr/lib/uucp`.

Existen otros archivos de configuración UUCP, tales como el `Dialers`, en donde se guarda información para comunicarse con modems, pero como en nuestro trabajo sólo utilizamos los 3 archivos principales, no nos referiremos más que a ellos.

Lo primero que debemos hacer es investigar los nombres de nodo uucp de nuestras máquinas. Estos nombres pueden encontrarse si se utiliza el comando `uname -i`. El nombre de la Sun es "espo", y el nombre de la U6000 es "Unisys".

Esto es muy importante de conocer, pues estos nombres son los que se emplean en la conexión UUCP, y son los mismos que se utilizarán dentro de los diferentes archivos de configuración para que se identifiquen las máquinas entre sí.

Para poder editar los archivos hay que tener privilegios de superusuario. Se puede usar cualquier editor de texto, pero nosotros trabajamos con el `vi` del Unix. La versión BNU viene con un programa de instalación amigable que se puede utilizar; nos referiremos detalladamente a esta opción más adelante.

Tenga presente que en estos archivos se cumple lo siguiente:

- 1- Las líneas que comienzan con `#` son comentarios.
- 2- Las líneas cuyo primer carácter es un espacio en blanco serán ignoradas.
- 3- Las líneas en blanco son ignoradas.
- 4- Los campos no utilizados en los archivos, deben llevar un guión.
- 5- El `\` es utilizado para unir 2 líneas.

2.4.1 Editando el archivo Systems.-

El archivo `Systems` consta de seis campos en los que se encuentra información referente a con que sistemas es posible conectarse, en que horario es posible establecer la comunicación, el nombre del dispositivo a usar para establecer

dicha comunicación, con que velocidad se la establecerá, de existir un número de teléfono, el login name y password del user de trabajo con que se conectará la máquina remota. Todo esto está detallado, tomando como ejemplo la línea digitada en el archivo **Systems** de la Sun:

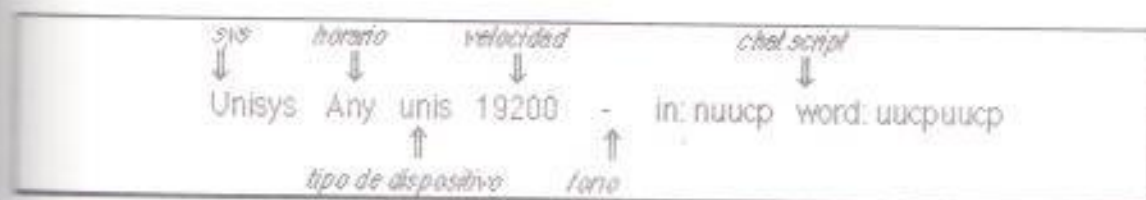


fig 2.4 Edición del archivo **Systems** de la Sun

- sys* El nombre de nodo del sistema remoto. Debe ser único con respecto a otros nombres de sistemas en este archivo.
- horario* Indica cuando el sistema local puede llamar al remoto. Any significa cualquier día y a cualquier hora. Existen opciones como Never y Wk para indicar un horario restringido para la conexión.
- tipo de dispositivo* El nombre del dispositivo para hacer la llamada. Esto es solo un puntero para el archivo **Devices**, y debe coincidir con su 1^{er} campo.
- velocidad* La velocidad del dispositivo en bits por segundo.
- tono* El número a marcar que empleará un modem para llamar. No usado en nuestro caso.
- chat script* Una cadena de caracteres describiendo la conversación inicial entre las dos máquinas. Describe el proceso completo de conexión. Indica el login name y el password del usuario de trabajo.

Además de la línea digitada en la Sun, se escribió la siguiente línea en el archivo **Systems** de la U6000.

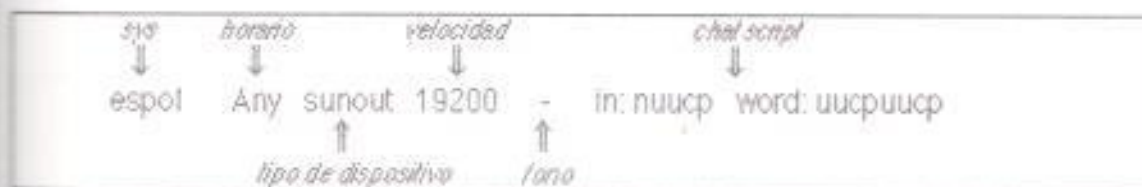


fig 2.5 Edición del archivo **Systems** de la U6000

***IMPORTANTE:** Nótese que en los archivos **Systems**, tanto de la Sun como de la U6000, el campo velocidad debe coincidir. Además en nuestro caso, esa es la velocidad a la que deben estar inicializados también los TAU's y los archivos **tytab** e **initab** de la Sun y la U6000 respectivamente.

2.4.2 Editando el archivo Devices.

Este archivo contiene información para realizar conexiones usando ACU's (automatic calling units), directas, o de red.

Cada entrada en este archivo debe comenzar en la columna 1. De otra manera se obtendrá un mensaje de error "NO DEVICES AVAILABLE" al tratar de usar ese dispositivo. Cada entrada en este archivo sigue el formato general de este ejemplo, en el cual mostramos la entrada editada en la Sun:



fig 2.6 Edición del archivo **Devices** de la Sun

tipo El campo *tipo* es apuntada por el campo *tipo de dispositivo* del archivo **Systems**. El campo *tipo* describe el tipo de conexión a realizar, y puede llevar una de las siguientes palabras:

ACU Para enlaces vía modem

Direct Para un enlace directo, vía modem o switch LAN a ser usado solo con el comando **cu**.

network Para enlaces hechos a través de una LAN.

sysname Para enlaces directos con una máquina particular. Puede ir cualquier nombre (este es el que utilizamos nosotros).

puerto de datos Es el nombre del dispositivo serial usado en la comunicación. Para enlaces directos y modems, este campo contiene el nombre del archivo especial en **ldev** que corresponde al puerto.

puerto dialer Es un campo opcional usado si el campo *tipo* es **ACU** y se usa para la llamada telefónica un código de área.

velocidad Es la velocidad que emplea el dispositivo que realiza la conexión. Se puede usar la clave **Any**, en cuyo caso se asume el campo *velocidad* del archivo **Systems**.

par dialer-token Contiene pares de nombres de "dialers" contenidos en el archi-

vo UUCP de igual nombre (ej: hayes, penril, ventel, 801, direct), y "tokens". Cada par representa un dialer y un argumento a pasar al dialer. Nosotros empleamos **direct** en nuestro enlace.

La entrada que nosotros editamos en el archivo **Devices** de la U6000 luce

asi:

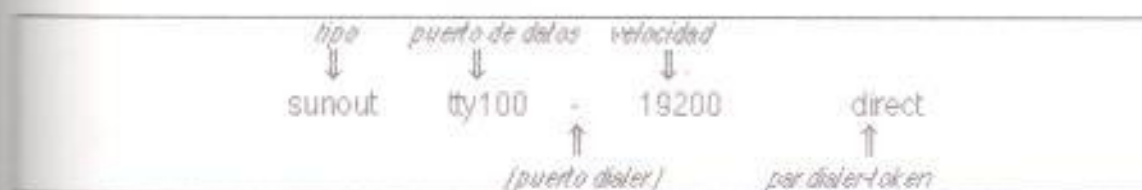


fig 2.6 Edición del archivo **Devices** de la U6000

2.4.3 Editando el archivo Permissions.-

El archivo **Permissions** es utilizado para implementar la seguridad de los sistemas en una conexión UUCP. Por ejemplo, limitar el número de comandos que una máquina remota conectada a Ud. puede ejecutar en su sistema, o limitar los archivos que se pueden acceder, etc.

Estos permisos son los que se concederán a los sistemas remotos, es decir, mi sistema local no se puede "dar" permisos en un sistema remoto, sino que el administrador del sistema remoto es el que le asignará los permisos a mi máquina dentro de su máquina, y viceversa.

Pero hay que señalar que los permisos UUCP no pueden pasar por encima de los permisos Unix que tenga asignado el sistema. Esto es, se emplean los permisos asignados al usuario Unix dentro de un sistema, que usa la conexión UUCP.

A continuación mostramos una lista de las palabras claves y sus significados:

MACHINE : Especifica las máquinas a las que el sistema local puede llamar.

LOGNAME : Especifica los login id's que pueden ser usados por localidades remotas para conectarse en nuestro sistema.

REQUEST : Especifica si el sistema remoto puede hacer pedidos de transferencia de archivos para su computador o no. El default es **no**. Se digito **yes** en ambos sistemas, permitiendoles hacer los pedidos de transferencias.

SENDFILES : Si ponemos **yes** indica que el sistema remoto puede enviar archivos a nuestro sistema cuando nosotros fuimos los que llamamos. Si se coloca **call**, el remoto debe de llamar para poder enviar archivos. El default es **call**.

READ : Especifica los directorios de los que uucico puede pedir archivos. El default es el **user/spool/uucppublic**. Si se pone **/**, se indica que se puede pedir cualquier archivo.

WRITE : Especifica los directorios en los que uucico puede escribir archivos. El default es el **user/spool/uucppublic**. Si se pone **/**, se indica que se puede escribir en cualquier archivo.

COMMANDS : Especifica una lista de comandos para los que el sistema remoto tiene permiso de ejecutar. El default es **rmail**.

Además existen otras opciones de permisos que no empleamos en el proyecto, tales como **VALIDATE**, **NOREAD**, **NOWRITE**, **CALLBACK**, **MYNAME** y **PUEDIR**.

A continuación escribiremos las líneas que editamos en los archivos de las máquinas. Para la Sun:

```
LOGNAME= nuucp      MACHINE= Unisys
READ=/      WRITE=/      COMMANDS= ALL
SENDFILES= yes    REQUEST= yes
```

fig 2.8 Edición del archivo **Permissions** de la Sun

Para la U6000:

```
LOGNAME= nuucp      MACHINE= esp01
READ= /      WRITE= /      COMMANDS= ALL
SENDFILES= yes      REQUEST= yes
```

fig 2.9 Edición del archivo **Permissions** de la U6000

2.5 Editando el archivo */etc/passwd* -

En los archivo */etc/passwd* se encuentran todos los login id's de los usuarios de un sistema Unix, indicando cosas como a qué grupo pertenece, si posee o no password, el directorio "home", etc.

Para una instalación UUCP es necesario tener un login id en este archivo, que es el que los uucico's usarán para identificarse durante una transferencia. Este login se conoce como un "login de trabajo", y es diferente del resto de usuarios. Este login usará como shell el */usr/lib/uucp/uucico*. Este shell solo funcionará para la conexión automática, la cual se realiza durante una transferencia UUCP.

En los sistemas Unisys ya viene creado al instalar el sistema operativo, no así en la Sun, donde hubo que crearlo. Adicionalmente se puede observar en este archivo el login id del administrador del uucp, como se ve en el siguiente ejemplo de lo que se escribió en la Sun (en la U6000 es muy similar):

```
password  groupid          home dir
  ↓        ↓              ↓
uucp : x : 8 : 5 : uucp administrator : /usr/lib/uucp
nuucp : x : 10 : 5 : uucp dial in : /usr/spool/uucppublic /usr/lib/uucp/uucico
  ↑        ↑        ↑              ↑
login id  user id  user id o          shell
```

fig 2.10 Edición del archivo */etc/passwd*

<i>login id</i>	uucp para el login del administrador, o algun nombre para login de trabajo, por ejemplo nuucp.
<i>password</i>	Palabra clave que se usa en el proceso de conexión. Por default viene sin password.
<i>user id</i>	Número que identifica a un usuario. Usualmente es 5 para uucp (administrador).
<i>group id</i>	Grupo al cual el usuario pertenece. Usualmente es 5 para grupo uucp.
<i>user info</i>	Etiqueta de comentario.
<i>home dir</i>	Indica el directorio "home" del usuario.
<i>shell</i>	Indica el entorno de trabajo del usuario. Debe ser /usr/lib/uucpl/uucico para login de trabajo. Para el administrador debe ser cualquier otro, aunque por default viene el /bin/sh .

Pueden existir varios login de trabajo, aunque esto no es necesario. En el archivo **Permissions** debe existir este login id.

2.6 Resumen.-

Toda la edición de los archivos, se efectuó en la Sun empleando el editor UNIX **vi**. En la U6000 se los puede editar manualmente con el **vi**, o empleando un menú amigable que viene con el sistema operativo.

De escoger esta segunda opción solo se puede entrar en el menú de administración siendo superusuario:

Una vez dentro del menú se escoje la opción **Network**; luego se escoje la opción **UUCP**, y por último, la opción **Set Up a line**.

Para trabajar con el menú, pese a ser amigable, se tienen que revisar mucho los diferentes manuales que vienen con el equipo Unisys, así como emplear los conocimientos de Unix (especialmente el editor **vi**).

Con el menú se pueden configurar distintos parámetros, como son :

selección de un puerto serial (ej: tty101)

disciplina

baudios

bits

paridad

control de flujo de salida

Una vez elegida la disciplina, sale un menú para bautizar al sistema remoto, escojer con cual tipo de conexión lo enlazaremos, que protocolo usaremos, un nombre del dispositivo usado, si será bidireccional o solo permitirá llamadas que entren o que salgan, login name, password, dialcode, teléfono, y restricciones de tiempo si las hay.

Luego se pasa a un menú para inicializar el archivo **Devices**, donde se elige el tipo de modem y se hace la configuración adecuada.

De aquí se pasa al menú para editar **Permissions**, que es el archivo de seguridad, donde especificaremos restricciones sobre la transferencia.

Finalmente podemos resumir los pasos para habilitar la conexión UUCP en:

- 1.- Seleccionar el tipo de conexión a utilizar.
- 2.- Seleccionar un puerto físico en las máquinas.
- 3.- Construir los cables necesarios.
- 4.- Setear los parámetros de los puertos (editarlos en el `inittab` ó el `ltytab`).
- 5.- Editar los archivos de configuración uucp (`Systems`, `Devices`, `Permissions`, etc.)
- 6.- Editar el archivo `letc/passwd`.

CAPITULO 3

MANUAL DEL USUARIO

Este capítulo está enfocado a facilitar el uso de la conexión UUCP instalada, para que los futuros usuarios tengan una buena referencia de como utilizar los comandos uucp y puedan beneficiarse íntegramente de todas las ventajas que ofrece este paquete de programas.

Aunque mostraremos una serie de procedimientos a seguir, el usuario no necesariamente tiene que regirse a ellos de una manera estricta si no, que a medida que vaya practicando el uso de los comandos, y llegue a un mejor entendimiento de ellos, podrá ir creando sus propios mecanismos a seguir, que se adapten mejor a sus necesidades individuales.

Es por esto que recomendamos leer este capítulo y practicar los procedimientos expuestos, así como ayudarse por medio del anexo A, el cual ofrece una guía rápida de los comandos y sus distintas opciones.

Para el correcto entendimiento de este manual se ha utilizado la siguiente convención: Las letras **en negritas** son las que el usuario debe escribir y las letras normales son las respuestas del sistema operativo. Las letras *en itálicas* son comentarios de este manual. El prompt de la U6000 utilizado en los ejemplos será **uucp\$**, y el de la Sun **espol.edu.ec%**.

Para comenzar la sesión UUCP, es necesario que el usuario tenga acceso (poseer un login id) a cualquiera de las dos máquinas de la conexión. Una vez que el usuario se haya conectado en una de las máquinas, ya se halla listo para realizar cualquiera de las tareas descritas a continuación:

3.1 Trabajando con el comando cu.-

Para poder emplear este comando, el usuario deberá tener acceso a las dos máquinas, esto es tener un login id en ambos sistemas operativos. Hay que tener presente que cuando se hace uso de este comando, se está bloqueando el acceso a la línea de comunicación para otros usuarios, por lo que es nuestra sugerencia el tratar de usar este comando con la mayor discreción posible.

3.1.1 Conectándose con el comando cu.-

El comando **cu** en nuestra conexión quedó restringido a la máquina Unisys. Es decir, solo se podrá utilizar para conectarnos desde la U6000 a la Sun. Una vez que nos encontremos conectados en la U6000 y suponiendo que existe un usuario de la Sun con login name **usuario** y password **itsme**, para conectarse con el comando **cu** se empleará el siguiente procedimiento:

```
uucp$cu -e espol
Connected
espol.edu.ec login: usuario          «← escriba su login name
password: itsme                       «← escriba su password (no se verá)
SunOs version.....                 «← mensaje de conexión de la Sun
no match...
espol.edu.ec%
```

En este momento el usuario **usuario** se ha conectado con la Sun y se encuentra en capacidad de efectuar cualquiera de los comandos que esta reconozca, tales como *who, date, cal, ls, cat, vi, ftp, telnet, cd, etc.*

Otra acotación de importancia es que no siempre se podrá emplear este comando, pues si otro usuario esta realizando alguna transferencia UUCP, estaria bloqueada la línea, y el sistema nos enviaria el siguiente mensaje: "DEVICE LOCKED".

3.1.2 Secuencias de escape tilde (~).

La ~ (tilde) permite escapar hacia un shell interactivo para poder realizar una serie de tareas. Por ejemplo, si digitamos :

```
espol.edu.ec% ~ [Unisys] !
```

```
uucp$
```

Al haber digitado la tilde seguido del signo de admiración hemos retornado a la U6000, sin desconectarnos de la Sun. Esto es muy útil por ejemplo si queremos ver qué archivos se encuentran en el directorio corriente de la Unisys, sin cortar la comunicación.

Para retornar a la Sun digite :

```
uucp$ <control-d>
```

```
espol.edu.ec%
```

Una vez que Ud. decida que ya no tiene ninguna tarea que realizar conectado en la Sun, puede cortar la comunicación con otra secuencia de escape tilde (~.), como se verá en el siguiente ejemplo:

```
espol.edu.ec% ~ [Unisys] .
```

```
Disconnected
```

```
uucp$
```

Pasaremos a explicar unas secuencia de escape con la tilde que nos permiten realizar transferencias de archivos. El resto de secuencias de escape con la tilde se encuentran en el anexo A.

3.1.3 Transfiriendo archivos con cu.-

Una vez conectados en la Sun, el comando **cu** nos permite bajar archivos desde la SunOS a la U6000 o enviarlos desde la U6000 a la Sun.

Supongamos que en la Sun tenemos un archivo con nombre **file1** y deseamos bajarlo a la Unisys; Lo primero que se debe hacer después de conectarse con la Sun es colocarse en el directorio donde se encuentra ese archivo. La secuencia de escape con la tilde empleada para tomar este archivo sería:

```
espol.edu.ec%~ [Unisys] %take file1 file2
```

La máquina enviará aquí mensajes sobre la transferencia realizada.

El archivo será bajado de la Sun y escrito en la U6000, con el nombre **file2**, dentro del directorio en el que se encontraba el usuario Unisys antes de digitar el comando **cu**. De no haber especificado un nuevo nombre, el archivo hubiera sido copiado con el mismo nombre (**file1**).

Ahora supongamos que queremos devolver este archivo con ciertos cambios y con el nombre **file2modif**. La nueva secuencia de tilde para realizar esto será:

```
espol.edu.ec%~ [Unisys] %put file2 file2modif
```

La máquina enviará aquí mensajes sobre la transferencia realizada.

3.2 Trabajando con el comando mail.-

El comando **mail** trabaja básicamente como su versión típica de UNIX. Nuestra implementación del UUCP quedó establecida de modo que solo podemos enviar correo desde la U6000 hacia cualquier usuario de la Sun.

Supongamos que estamos conectados en la U6000 y queremos enviar un mail al usuario **usuario** de la Sun, el formato a seguir es el siguiente:

```
uucp$ mail espoff!usuario
estoy probando la conexión UUCP
enviame un archivo con el comando uucp
si recibiste este mail
saludos,
-
uucp$
```

Es de anotar que las opciones y servicios de mail son las mismas que en el mail UNIX común y corriente. Por ejemplo el usuario **usuario**, recibirá en la Sun un mensaje:

```
you have mail
```

3.3 Trabajando con los comandos de transferencia de archivos.-

El paquete UUCP proporciona una gran versatilidad para transferir archivos, pudiendo bajarlos o enviarlos al directorio público o a cualquier otro directorio (dependiendo de los permisos que se tengan).

Existen dos comandos para transferir archivos, el comando **uuto** y el comando **uucp**. El comando **uucp** permite realizar la transferencia en los dos sentidos, esto es enviar y recibir archivos, desde y hacia cualquier directorio de los sistemas involucrados. El comando **uuto** en cambio, sólo permite enviar archivos desde un directorio cualquiera de alguna de las dos máquinas, hacia el directorio público de la

receptora, de donde deberán ser recuperados usando el comando **uupick** por parte de los usuarios a los que el **uuto** les avise con un correo.

3.3.1 Los comandos uuto y uupick.-

Supongamos que nos encontramos conectados con la Unisys y queremos enviar su archivo **file2** al usuario **usuario1** de la Sun (sistema **espol**) ; primero que nada nos colocamos en el directorio en que se encuentra dicho archivo; luego **digitamos**:

```
uucp$ uuto file2 espol!usuario1
```

```
uucp$
```

Cuando el archivo le llegue a la Sun, puede ser encontrado en el PUBDIR dentro del siguiente directorio:

```
/usr/spool/uucppublic/recv/usuario1/Unisys/file2
```

El *recv* es un subdirectorio del PUBDIR que es usado por **uuto**. Además el usuario **usuario1** será notificado por correo de la llegada del archivo y su procedencia.

Existen otras opciones de **uuto** que pueden emplearse (refierase al anexo A) en la transferencia.

El usuario **usuario1** puede copiar o mover este archivo del directorio público a cualquier directorio del sistema al que él tenga permiso de escritura. Puede hacer esto directamente con los comandos Unix **mv** o **cp** o empleando el comando UUCP **uupick**.

Una vez que el usuario **usuario1** recibió un mail confirmándole de la llegada del archivo **file2**, podría hacer lo siguiente:

```
espol.edu.ec% uupick
remote from system Unisys: file file2
?
m <RETURN>
7 blocks
espol.edu.ec%
```

En este ejemplo se usó la opción **m** del ambiente **uupick**, la cual nos permite mover el archivo al directorio en el que se encontraba antes de digitar **uupick**.

Dentro del ambiente **uupick** se podrían realizar otras tareas, cuyas opciones se hallan listadas en el anexo A.

3.3.2 El comando uucp.-

Supongamos que tenemos un archivo **file2** dentro de la U6000 y queremos enviarlo hacia el directorio público de la Sun. Primero que nada nos conectamos en la Unisys y nos colocamos en el directorio donde se encuentra **file2**. Luego procedemos de la siguiente manera:

```
uucp$ uucp file2 espol//usr/spool/uucppublic/file3
```

El archivo **file2** será copiado en el PUBDIR de la SunOS con el nombre **file3**. De haber omitido el nombre en el camino (path) especificado en el comando, se hubiera copiado con el mismo nombre, esto es **file2**.

También se pudo haber digitado:

```
uucp$ uucp file2 espoll~/file3
```

2.4 Que realizará la misma función que el comando anterior, pues la tilde es una abreviación del directorio público.

Una gran ventaja del comando **uucp** sobre el **uuto** es, como ya se indicó anteriormente, que se puede enviar archivos hacia cualquier directorio, y no solo al PUBDIR, por ejemplo, quiero enviar el mismo archivo pero ahora al directorio **/home/topico/usuario1** de la SunOS, para lo cual digito:

```
uucp$ uucp file2 espoll!/home/topico/usuario1/file3
```

Ahora supongamos que se quiere bajar de la SunOS a la U6000 el archivo **file3** del ejemplo anterior a mi directorio de trabajo corriente, con el mismo nombre. Entonces se digitará:

```
2.5 uucp$ uucp espoll!/home/topico/usuario1/file3
```

Si por ejemplo hubiera querido bajarlo no a mi directorio de trabajo sino a otro directorio cualquiera (en este caso el **/etc/bin**), y con el nombre **file4**, se hubiera escrito:

```
uucp$ uucp file2 espoll!/home/topico/usuario1/file3 /etc/bin/file4
```


3.4 El comando Uutry

Este comando invoca al demonio **uucico** manualmente efectuando de esta forma, todos los trabajos encolados.

El comando **Uutry** se empleará en nuestra conexión para recuperar trabajos encolados de la Sun. Para utilizarlo basta con ubicarse en el directorio **/usr/lib/uucp** y digitar:

```
uucp% Uutry espol
```

Esto se debe a que la Sun no puede llamar a la U6000, pues las limitaciones de los sistemas obligaron a implementar la línea unidireccionalmente.

3.5 Procedimientos recomendados.-

Debido a que se pidió que la conexión permita dar salida a red Internet, es decir principalmente bajar y enviar archivos y correos, recomendamos:

Para enviar correo, primero conéctese a la U6000, luego conéctese en la Sun con **cu** y trabaje como se hace normalmente en la red Internet. Para la recepción de correo, conéctese con **cu** a la Sun, grabe los archivos de correo que su usuario de la SunOS haya recibido de la red Internet en un archivo cualquiera, y proceda a bajarlos con cualquiera de los comandos de transferencia de archivos UUCP ya vistos.

Para bajar archivos de la red Internet primero conéctese con **cu** en la Sun y utilice el comando **ftp** como en cualquier sesión de transferencia TCP/IP, y luego de bajar los archivos requeridos a la SunOS, transfiera a la U6000 con cualquier

comando de transferencia de archivos UUCP. Puede ser necesario invocar el **Uutry** en la U6000 para recibirlos.

Para enviar archivos a la red Internet, envíelos a la Sun con cualquiera de los comandos de transferencia de archivos, luego conéctese en la Sun con **cu** y trate esos archivos como en cualquier sesión de transferencia TCP/IP.

CONCLUSIONES Y RECOMENDACIONES

El uso de una red de comunicaciones proporciona una gran conveniencia para cualquier persona. Permite a la gente de diferentes localidades intercambiar correo o archivos, tal y como usuarios de un mismo sistema lo hacen. Permite además compartir recursos, como por ejemplo un banco de impresoras.

El UUCP es un paquete de programas que da una facilidad de red para los sistemas operativos UNIX. Su software consiste de archivos y programas de configuración y administración, y un número de programas que dan a los usuarios acceso a estos. En resumen, el UUCP proporciona al usuario UNIX:

- Envío de correo a usuarios en sistemas remotos.
- Transferencia de archivos entre sistemas UNIX.
- Acceso a recursos de un sistema remoto.

La instalación del UUCP puede presentar otra ventaja futura, que sería la incorporación de la ESPOL a la red Usenet, la cual es una red de noticias que emplea el software de dominio público llamado Netnews para intercambiar "noticias" sobre una amplia variedad de tópicos. Usenet incluye a miles de sistemas, muchos de los cuales utilizan UUCP como su protocolo básico de red. Además Usenet está conectada a través de "Gateways" a muchas redes de computadoras, como la Internet.

El paquete de programas de dominio público Netnews incluye software que permite a los usuarios leer y enviar noticias a la red. Además del intercambio de noticias, el conectarse a la red Usenet permite a los usuarios comunicarse vía **mail** con usuarios de otras localidades conectadas a la red Usenet.

Sin embargo, debido al volumen de tráfico involucrado, se necesita de una gran cantidad de espacio libre en el disco. Este tráfico puede llegar a ser de hasta 4 MByte por día. Ya que los artículos se dejan en la red usualmente por una o dos

semanas para permitir a los usuarios leerlos, se necesita de 28 a 56 MBytes de espacio de disco libre. Además, en el caso de enlazarnos a la red Usenet, se requiere la capacitación de un administrador para esta red. Por esta razón los integrantes de este grupo de trabajo no recomendamos que en la actualidad se realice un enlace con la red Usenet. Si en un futuro la ESPOL se decide a enlazarse con Usenet, lo primero que debe hacer es ampliar la capacidad de los discos de sus sistemas y tener presente que se debe dejar una gran cantidad de esa memoria reservada.

Hay una infinidad de ejemplos en donde se puede obtener un máximo beneficio del UUCP, veamos tres de ellos:

1. En un pequeño negocio donde hay varios sistemas de minicomputadores, pero sólo una impresora laser, se podría usar UUCP para enviar a imprimir archivos que los usuarios de estos minicomputadores necesiten.
2. Una cadena de pequeños negocios tiene un sistema UNIX en cada tienda para recolectar datos de cada caja registradora. Se puede usar el UUCP para que por la noche transfiera los archivos que contengan la información sobre las ventas del día a un gran computador UNIX en la oficina central, donde esta información será procesada.
3. Varios programadores se encuentran trabajando en distintas localidades, en un solo gran proyecto. Los miembros de este equipo pueden emplear el UUCP para intercambiar archivos actualizados, así como para enviarse mensajes entre ellos en el transcurso de su trabajo.

Como se ve, el UUCP requiere de algún grado de cooperación de los sistemas remotos y sus usuarios.

Es importante hacer notar que no fue posible habilitar el mail desde la Sun hacia la U6000 debido a que las direcciones que usa la SunOS para reconocer usuarios han sido cambiadas para TCP/IP, el cual es ampliamente utilizado en la ESPOL.

Como conclusión final de este trabajo podemos agregar que el paquete de programas UUCP es un poderoso mecanismo para realizar transferencias de archivos entre sistemas Unix, y además proporciona otras facilidades como un correo electrónico y un comando para conectarnos en una máquina remota como un terminal.

Los integrantes de este grupo de trabajo consideramos que pese a los beneficios que nos brinda el UUCP, este no se ajusta a todas las necesidades de interconexión con la red Internet, es decir, las expectativas para las que se diseñó este proyecto, no se cumplen a cabalidad, por limitaciones inherentes al UUCP y, por modificaciones en el kernel de la SunOS. Por lo anteriormente anotado, es recomendación de este grupo el realizar la conexión de la U6000 con la red Internet utilizando el protocolo TCP/IP, para lo cual se necesitaría solamente instalar una tarjeta Ethernet en el server Unisys, puesto que el software en sí ya se encuentra instalado en él.

Durante el desarrollo de éste proyecto, los integrantes de este grupo de investigación nos vimos en la necesidad de estudiar el sistema operativo UNIX y su manejo, tanto para la U6000 como para la SunOS, pues se encontraron sustanciales diferencias en sus versiones UNIX. Además, los problemas encontrados durante la instalación nos obligaron a investigar más a fondo los conceptos sobre comunicaciones. Se tuvo que afrontar problemas tales como fallas en la inicialización de los puertos, daños en la línea dedicada, mal funcionamiento de los TAU's, la caída del sistema operativo de la U6000, etc. Todas estas vicisitudes nos han mostrado cómo en la práctica los Ingenieros deben recurrir a sus recursos, no solo de conocimientos, sino de relaciones humanas y sentido común para lograr sus

objetivos. Hemos aprendido a relacionarnos con la gente y obtener su cooperación. Hemos aprendido a leer manuales técnicos. Hemos aprendido a trabajar con herramientas. En resumidas cuentas, el desarrollo de este proyecto nos ha servido para completar nuestra formación académica y de esta manera lanzarnos con seguridad al ejercicio de nuestra profesión.

BIBLIOGRAFIA

- Tim O'Reilly & Grace Todino, **MANAGING UUCP AND USENET**. O'Reilly & Associates, Inc.; Sebastopol-California, Estados Unidos; 1992, Décima Edición.

- Dale Dougherty & Grace Todino, **USING UUCP AND USENET**. O'Reilly & Associates, Inc.; Sebastopol-California, Estados Unidos; 1991, Décima Edición.

- Mitchell Waite, Stephen Prata, Donald Martin, **INTRODUCCION AL UNIX SISTEMA V**. Editorial Anaya, Madrid-España; 1989, Segunda Edición.

- Gilbert Held, **UNDERSTANDING DATA COMMUNICATIONS**. Sams, Carmel-Indiana, Estados Unidos; 1992, Tercera Edición.

- Manuales de la U6000- Unisys:

Administration Guide.

Operations Guide.

Administration & Operations Guide.

Error messages manual.

User's Reference manual Vol. No. 1,2,3.

Programmers Reference manual Vol. No. 1,2,3.

Administration Reference manual Vol. No. 1,2,3.

- Manual de la SunOS-Sun:

System Administrator Guide.