

MNX Messenger
Memoria del Proyecto de Ampliación de Sistemas Operativos
Grupo 2º I.T. en Informática de Gestión, curso 2006-2007

➤ **Introducción:**

Andrew Stuart "Andy" Tanenbaum crea en 1987 el sistema operativo de micronúcleo Minix, una réplica gratuita de UNIX con propósitos educativos, que posteriormente fue la inspiración de Linux. Se distribuye junto con su código fuente y existe amplia documentación, pero debido a su orientación educativa se ha mantenido muy simple y carece de muchas características.

Pensamos que una de las cosas más prácticas que se le podrían añadir sería un sistema de mensajería instantánea tipo Messenger. Además nos parece una opción interesante porque aparte de tratar el tema de los terminales y el control de sesiones, lo podíamos complicar tanto como quisiéramos.

Hemos creado un programa que no solo implementa las características propuestas si no que añade varias más, aunque alguna se ha desechado y otras se realizan de manera distinta, se detalla posteriormente.

➤ **Objetivo:**

Añadirle al SO Minix una herramienta de mensajería tipo Messenger.

➤ **Fundamentos teóricos:**

Minix, como ya sabemos y al igual que el resto de sistemas UNIX, mantiene en el archivo `/etc/passwd` una lista de todos los usuarios registrados. Existe una función C en las librerías estándar de estos sistemas, `getpwnam(char *login)`; que nos devuelve una estructura tipo `passwd` para consultarla fácilmente.

Pero para saber quién está conectado al sistema actualmente, mantiene otra lista en el archivo `/etc/utmp`, que es al que acceden por ejemplo los comandos `last`, `who` y `w`. Leemos registros tipo `utmp` y comparamos el campo `ut_name` con el `login` que queremos comprobar. Si lo encontramos, sabemos que está conectado y además tenemos en el campo `ut_line` el nombre del terminal (`tty`) en el que está conectado (aparecerá tantas veces en la lista como distintas conexiones tenga).

Por otra parte, Minix maneja los distintos terminales a los que se conectan los usuarios como un fichero más, al igual que los dispositivos, abstrayéndonos del hardware. Estos se encuentran dentro del directorio `/dev/`, y sabiendo el nombre del terminal concreto, podremos abrirlo con funciones normales de ficheros y escribir en él como si de un fichero se tratase. Esto tiene el efecto de escribir con el teclado sobre el terminal.

➤ Descripción del trabajo realizado:

Hemos empezado creando la estructura del programa general, los menús y submenús, y las funciones para insertar, eliminar y recorrer una lista enlazada para los contactos, y guardarla y cargarla en el disco, y también las opciones y el historial.

La idea inicial era que se avisara de los inicios y cierres de sesión del sistema Minix, y no tuvimos problemas para implementarlo, pero consideramos más apropiado que se den al iniciar y cerrar el programa de mensajería. Eso sí, se controla si nuestros contactos han iniciado sesión en Minix para poder enviarles mensajes, usando la información comentada en fundamentos teóricos.

Para entender como funciona el envío de mensajes, consultamos el código fuente del comando `write`, disponible para el sistema Minix. Con él entendimos cómo tratan los sistemas UNIX los terminales, y lo implementamos también de la forma descrita en el apartado anterior. También preparamos una función para que si se recibe una señal de interrupción (por ejemplo `<Ctrl.+C>`) se vuelva limpiamente a nuestro terminal.

Después fuimos añadiendo algunas características más, como la de enviar mensajes generales, poner apodos, guardar un historial, eliminar contactos, modo silencioso, enviar mensajes con pitidos, etc. y nos documentamos también de los caracteres de escape para mejorar la presentación, borrando la pantalla y situando el cursor en la posición que quisiéramos.

➤ Conclusiones:

Hemos conseguido el objetivo general planteado, dotar al sistema Minix de un sistema de mensajería tipo Messenger, siendo además lo más fácil de utilizar posible, y permitiendo varias opciones adicionales.

Creemos que es una buena aportación para Minix pues está clara la gran popularidad que tienen estos programas para los usuarios.

Además hemos ampliado los conocimientos de Minix vistos en teoría con los nuevos aspectos de este proyecto, básicamente los terminales y las sesiones de usuario.

➤ Propuestas de ampliación:

Se podrían añadir más comandos, por ejemplo uno que permitiese ejecución de programas en el terminal remoto, ó tratar de controlar la entrada de mensajes que se realizan sobre nuestro terminal (no solo la salida) pero creemos que esto complicaría demasiado el proyecto.

➤ Bibliografía:

- Molay, Bruce, *Understanding Unix/Linux Programming: A Guide to Theory and Practice*, Ed. Prentice Hall, primera edición, 2003
- <http://www.minix3.org/manpages/man1/write.1.html>
(Última consulta: 23/04/2007)
- <http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node91.html>
(Última consulta: 18/05/2007)

➤ Anexo I. Trabajo en grupo:

➤ Anexo II. Manual de usuario:

Instalación:

Basta con el ejecutable, él crea los archivos de configuración, contactos e historial necesarios, pero cada usuario que lo quiera utilizar debe tener preferiblemente su propia copia en una carpeta distinta.

Ejecución:

Al iniciar el programa aparece el menú principal con todas las opciones:

```
-----  
Puck - MNX Messenger ©  
-----  
  
1) Iniciar una conversacion  
2) Enviar un aviso general  
3) Agregar un contacto  
4) Eliminar un contacto  
5) Mostrar historial  
6) Configuracion  
7) Cerrar sesion  
  
Escriba la opcion deseada: █  
-----
```

1) Iniciar una conversación

Se mostrará una lista con nuestros contactos, indicando si están conectados o no (deben conectarse a la misma máquina Minix con <mlogin>). Introducimos un número de usuario, y si está conectado podemos empezar a mandarle mensajes. Para terminar escribimos exit y pulsamos <Enter>. También podemos utilizar algún comando, como “beep” para enviarle un pitido:

```
Puck - MNX Messenger ©
```

```
Lista de contactos:
```

N. Nombre	Nick	Estado
1) raudela	Raul	Desconectado
2) ricvare	Ricardo	Conectado
3) jospint	Jospint	Desconectado
4) artruíz	Arturo	Desconectado

```
- Introduzca numero de contacto: 2
```

```
Escriba texto, pulse <Enter> para enviar y escriba "exit" para terminar:  
>Hola
```

2) Enviar un aviso general

```
Puck - MNX Messenger ©
```

```
Aviso general a todos los contactos:
```

```
- Escriba el mensaje a enviar: Silencio!
```

Con esta opción, el mensaje que enviaremos se difundirá a todos nuestros contactos. También se puede usar el comando beep.

3) Agregar un contacto

```
Puck - MNX Messenger ©
```

```
Agregar un contacto:
```

```
- Escribe el login del contacto: davielop  
- Escribe el nick del contacto: David
```

Se nos pedirá el login y el nick del nuevo contacto y se añadirá a la lista.

4) Eliminar un contacto

Opción para eliminar un contacto. Se nos mostrará la lista y se nos pedirá el número de contacto para eliminarlo:

Puck - MNX Messenger ©

Lista de contactos:

N. Nombre	Nick	Estado
1) raudela	Raul	Desconectado
2) ricvare	Ricardo	Conectado
3) jospint	Jospint	Desconectado
4) artruiiz	Arturo	Desconectado
5) davielop	David	Desconectado

- Introduzca numero de contacto: █

5) Mostrar historial

Puck - MNX Messenger ©

Historial de mensajes enviados:

Wed May 23 04:05:37 2007
Mensaje de Puck a ricvare:
Hola

Wed May 23 04:05:46 2007
Mensaje general de Puck:
Aviso general!!

Pulse <Enter> para continuar . . . █

Con esta opción se nos mostrará el historial completo de mensajes enviados, tanto individuales como generales.

6) Configuración

Puck - MNX Messenger ©

Configuracion:

- Introduzca su nick: Puck
- Guardar un historial de mensajes (S/N)? s
- Iniciar/cerrar sesion en modo silencioso (S/N)? n █

En esta opción configuraremos el programa. Tenemos tres opciones:

- Introducir nuestro nick, el sobrenombre con el que nos verán el resto de usuarios. Por defecto es "Anónimo".
- Guardar un historial de mensajes: se registran los mensajes mientras esté activada. Por defecto está desactivada.
- Iniciar/cerrar sesión en modo silencioso: si está activada, al iniciar y cerrar sesión no avisaremos al resto de contactos. Por defecto está desactivada.

7) Cerrar sesión

Se guardará la configuración actual y la lista de contactos, se avisará a todos de que cerramos sesión (si no estamos en modo silencioso) y termina el programa.