

CONTROL Y SUPERVISIÓN REMOTOS DE EXPERIENCIAS DE LABORATORIO.

N. Martínez, E. Spinelli, A. Veiga

Dto. de Física, Universidad Nacional de La Plata. CICpBA.
C.C 67 1900 La Plata Argentina.

e-mail: electro@venus.fisica.unlp.edu.ar

Se presenta una propuesta para la supervisión y control centralizado de experimentos, con capacidad de acceso remoto. Se basa en la utilización de una computadora tipo PC, soportando un sistema operativo multitareas (Linux) que le permita atender, por un lado a distintos periféricos, y por otro conectarse a una red de comunicaciones (Internet), utilizando protocolos standard (TCP/IP).

Los periféricos podrán estar conectados a la computadora directamente en alguno de sus slots EISA, PCI, a través de interfaces standard (RS-232, GPIB) o definidas por el usuario.

La elección del sistema operativo no es excluyente, pero debe considerarse que Linux es un sistema abierto, de dominio público, con adecuadas herramientas de programación e interfaz gráfica, y que permite operación en tiempo real. La función de la computadora central podrá ir desde la recolección de datos y el pasaje de parámetros, hasta el control del experimento.

I. INTRODUCCIÓN

Actualmente, es común encontrar en laboratorios experimentales, sistemas de adquisición asados en computadoras tipo PC (multicanales, multiescalímetros, etc.). En la mayoría de estos casos la computadora es solamente un soporte para el almacenamiento de datos y el software de interfaz con el usuario, desarrollando sólo una tarea a la vez.

Dado que (aun en el caso de las configuraciones más modestas) una computadora tipo PC es una herramienta muy poderosa y versátil con gran capacidad de procesamiento, si se la provee de un adecuado sistema operativo y programas de aplicación, su utilización dentro de un sistema de control de un experimento, puede extenderse mucho más allá del simple soporte de una placa de adquisición de datos pudiendo realizar el control y procesamiento de varias variables de una manera muy eficiente.

Otro de los objetivos de este proyecto es agregar conectividad a los laboratorios. Esto puede lograrse usando sistemas operativos que dispongan de herramientas standard para el manejo de redes de datos locales (e inclusive Internet). Por lo tanto se propone la utilización del soporte de redes de comunicaciones y acceso remoto (ya existentes, en general) para proveer acceso a la supervisión y control de los experimentos

II. DESCRIPCIÓN GENERAL

La arquitectura propuesta está basada en una PC, conectada a una red local de comunicaciones. Los instrumentos que componen el experimento se conectan a la PC mediante distintas interfaces y son manejados a través de drivers específicos para cada una de ellos. Un programa de aplicación se encarga del monitoreo y con-

trol del experimento y de proporcionar la presentación del mismo a la red.

Esta PC que puede estar configurada para operar sin monitor ni teclado, es accedida desde la red para su configuración y programación, inclusive puede considerarse la posibilidad de montarla en un gabinete tipo rack.

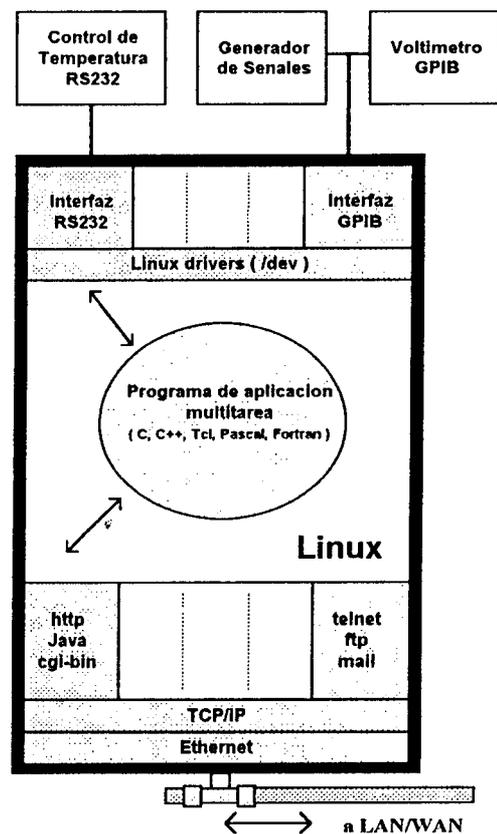


Figura 1

III. HARDWARE DEL SISTEMA

Instrumentación

Dado que se trata de organizar el funcionamiento de un experimento alrededor de una PC, todos los dispositivos deberán estar preparados para hacerlo. Ya es común que todos los instrumentos de uso general vengan provistos de esta facilidad a través de interfaces normalizadas (bus ISA, RS-232, GPIB, etc). También podrán diseñarse interfaces específicas que se conecten directamente al bus de la PC (ISA, EISA, PCI). En estos casos el driver correspondiente deberá ser escrito dentro de la filosofía de desarrollo planteada.

La computadora

Los requerimientos mínimos de la computadora no son muy exigentes, es suficiente con una PC con procesador 80386 o superior con 4MB de RAM, sin monitor ni teclado. Es importante notar que un sistema operativo como Linux (que es el elegido en nuestro caso) hace un aprovechamiento del hardware, mejor que lo que puede obtenerse utilizando DOS.

IV. SOFTWARE DEL SISTEMA

Sistema Operativo.

Tradicionalmente, y posiblemente por ser el sistema más fácil de implementar, se ha utilizado el sistema operativo DOS para el desarrollo de aplicaciones de laboratorio, en las cuales una PC de arquitectura standard tenía a su cargo alguna tarea específica del total de la experiencia. Este sistema operativo no ha evolucionado en conjunto con la tecnología que permite, como dijimos, disponer de potentes herramientas de cómputo a muy bajo costo. Por este motivo, diseñar una aplicación en DOS resulta dificultoso debido a la falta de herramientas de desarrollo.

Con el fin de obtener el máximo aprovechamiento de los recursos disponibles en una computadora tipo PC, en su utilización en la implementación de sistemas de adquisición de datos y control, hemos considerado la utilización de sistemas operativos de 'tiempo real' y Unix de propósitos generales. Estos proveen una gran variedad de herramientas que nos permiten explotar al máximo la capacidad de los nuevos sistemas de hardware.

Para esta aplicación se adoptó Linux. La elección del sistema operativo no es excluyente, pero debe considerarse que Linux es un sistema multitareas abierto, de dominio público, con adecuadas herramientas de programación, interfaz gráfica y conectividad y portabilidad (POSIX.1). Permite la operación en tiempo real, con algunas modificaciones y limitaciones. Un aspecto muy importante de este sistema operativo es la disponibilidad de las fuentes de todo el sistema, lo que facilita el diseño de nuevos drivers. Además utilizando hardware standard (ISA, GPIB, etc) los drivers se encuentran disponibles y son de dominio público.

Es importante destacar que Linux es un sistema abierto, que se encuentra en constante evolución, y al que hacen aportes un gran número de investigadores de todo el mundo.

Otros sistemas operativos están siendo analizados también, poniendo especial interés en su capacidad de manejar tareas de tiempo real y sistemas distribuidos. Si bien Unix no es la solución óptima para este tipo de sistemas si tienen requerimientos de tiempo muy estrictos. Existen sistemas operativos especialmente diseñados para estas aplicaciones (ej: QNX), pero se trata de productos comerciales de costo elevado.

Programa de Aplicación

La función del programa de aplicación podrá ir desde la recolección de datos y el pasaje de parámetros, hasta el control total del experimento. Está encargado también de presentar los datos para su acceso desde la red. En Linux existen diversas herramientas para implementar este programa como compiladores C, Pascal, y herramientas específicas para tareas gráficas como Xlib, Tcl/Tk, Motiff, etc. Para la utilización de las facilidades de la red local pueden utilizarse tanto protocolos standard (http, ftp, etc.), externos al programa, como las librerías de "sockets" para los programas en C.

La característica multitarea del Linux permite escribir programas que ejecuten tareas concurrentemente, de esta manera las distintas acciones que forman el monitoreo y supervisión del experimento pueden ser implementadas en rutinas independientes, dando como resultado un programa modular fácilmente interpretable.

Presentación del sistema en la red

Los protocolos utilizados para la presentación de los datos en Internet, como así también para la recepción de comandos, son los standard para un sistema operativo Unix. Esto permite operar el sistema con software que generalmente está disponible en las estaciones de trabajo conectadas a la red (Netscape, Java, ftp, telnet, etc.). Los protocolos de comunicación utilizados son el conjunto TCP/IP, que es nativo de los sistemas operativos Linux, lo que permite gran conectividad sobre la red local, incluso sobre Internet. El uso de protocolos standard facilita notablemente el desarrollo de aplicaciones que implementan el acceso a una red local.

El acceso remoto puede hacerse con el comando "telnet" para la operación remota y "ftp" para la transferencia de archivos. Las aplicaciones gráficas pueden ejecutarse en la PC del laboratorio y presentarse en la pantalla de otra PC conectada a la red utilizando las facilidades del sistema de X Windows. La programación de estas aplicaciones gráficas puede hacerse con una gran variedad de herramientas que se encuentran disponibles en la instalación standard del Linux. Están van desde C y C++ utilizando las librerías de X o Motiff, hasta lenguajes interpretados de más alto nivel como el Tcl/Tk, que

facilita mucho la implementación de GUIs (Graphical User Interfaces). Debe señalarse que estas herramientas son de dominio público y no requieren de licencias para ser utilizadas.

El acceso remoto puede realizarse también a través de herramientas standard de navegación por Internet (por ejemplo Netscape), instalando en la PC del laboratorio un servidor de HTTP (protocolo utilizado en el WWW). La nueva generación de estos servidores incluyen no solo la capacidad de transferir 'páginas', permiten además la ejecución remota de comandos en la PC del laboratorio, accediendo con Netscape. Esta facilidad se denomina CGI (Common Gateway Interface), y permite que un set de comandos o programas de aplicación definidos por el usuario puedan ser ejecutados remotamente desde una 'página'. De esta manera es posible presentar, por ejemplo, en una página accesible por Internet el panel de control de la experiencia.

Dado que el acceso remoto, se hace a través de una red pública, deberá implementarse un adecuado sistema de control de usuarios, utilizando el sistema de filtros del acceso de red, que está disponible en las nuevas versiones de Linux.

V. APLICACIÓN

A modo de ejemplo, se implementó un sistema de supervisión para una experiencia montada en el Laboratorio de Electrónica del Departamento de Física que involucra un equipo de control de temperatura con interfaz RS-232, y varios instrumentos con interfaz IEEE.488. Se modificaron los drivers de Linux para la puerta serie, se agregaron los correspondientes a la interfaz IEEE.488 y se escribió un programa en C que permite monitorear el estado de la experiencia.

Para el comando y supervisión remota se implementaron distintas alternativas. Para los instrumentos GPIB se utilizó una interfaz en modo texto utilizando el comando *telnet* (fig. 2) y una interfaz gráfica escrita en Tcl/Tk para ser utilizada en el entorno Xwindows (fig. 3).

```
[user@igi] show /dev/gpib7
+1.43463E+0
+1.43464E+0
+1.43464E+0
+1.43463E+0
+1.43465E+0
+1.43466E+0
+1.43467E+0
+1.43467E+0
+1.43466E+0
+1.43467E+0
+1.43466E+0
```

Figura 2

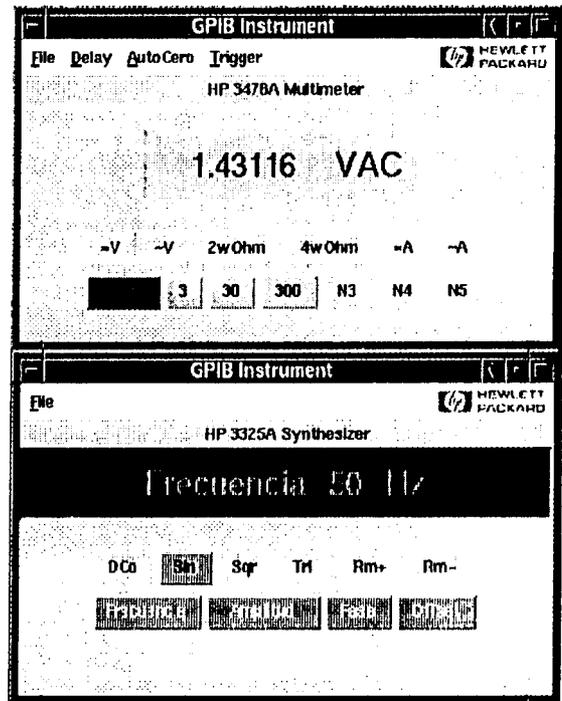


Figura 3

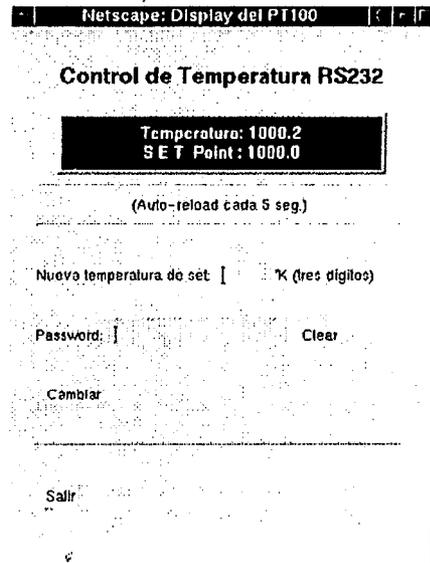


Figura 4

Para el control de temperatura, se implementó el acceso a través de un CGI que utiliza las facilidades del lenguaje Java, también disponibles en Linux. De esta manera, el monitoreo y comando remotos de este control puede hacerse utilizando Netscape o similar (fig. 4).

Una versión de demostración se encuentra en funcionamiento en <http://beba.fisica.unlp.edu.ar/>.

VI. CONCLUSIONES

Se presentaron los aspectos a tener en cuenta en el diseño de experiencias de laboratorio con capacidad de monitoreo y operación remota. La solución desarrollada, que se basa en la utilización de una computadora tipo PC, soportando un sistema operativo multitareas, invo-

lucra distintas elecciones y no pretende ser general. Es el resultado de experiencias realizadas en el Laboratorio de Electrónica del Departamento de Física de la U.N.L.P. buscando una implementación adaptada a las necesidades y posibilidades de los grupos de física experimental de este Departamento.

El sistema resulta útil tanto para el monitoreo de experiencias desde grandes distancias como para la supervisión desde una oficina ubicada en el mismo edificio y transferencia de datos del sistema de adquisición a los ordenadores de cálculo.

REFERENCIAS:

- 1- K. Hwang F. Briggs, Computer Architecture and Parallel Processing, Mc Graw Hill, 1984.
- 2- A. Tanenbaum, Modern Operating Systems, Prentice Hall, 1992.
- 3- W. Stevens, Advanced Programming in the Unix Environment, Addison Wesley, 1992.
- 4- Linux Lab Project Documentation, <http://www.redhat.com/HyperNews/get/Kho.html>
- 5- R. Baruch C. Schoreter, Writing Character Device Driver for Linux, Linux Lab Project Documentation, 1994.
- 6- Sunsite Linux Documentation Project, <http://sunsite.unc.edu/mdw/linux.html>
- 7- Olaf Kirch, Linux Network Administration, Linux Documentation Project, 1993.
- 8- Peter Baumann, The Linux Serial Programming HOWTO, <http://amelia.db.erau.edu/ldp/HOWTO>.
- 9- Michael Sweet, Serial Programming Guide for Compliant Operating Systems, <http://www.easysw.com/~mike/serial>, 1996.
- 10- M. Barabanov, V.Yodaiken, Introducing Real-Time Linux; Linux Journal, #34, Feb.1997.
- 11- Brent Welch, Practical Programming in Tcl and Tk, 1997. ISBN: 0-13-616830-2 1997.
- 12- David Martland, A brief introduction to Tcl/Tk, <http://http2.brunel.ac.uk/~csstddm/tcl2/>
- 13- QNX Software Systems Ltd., QNX Operating System, 1996.
- 14- C Shroeter, Linux-GPIB User's Guide, Clausi@chemie.fu-berlin.de., 1996.