

USO DE MINIX PARA LA ENSEÑANZA DE SISTEMAS OPERATIVOS

Martín SANTANA (*)
Miguel SANTANA (**)

La enseñanza en el área de los Sistemas Operativos siempre se ha visto confrontada a una serie de dificultades, que conducen en muchos casos a la realización de cursos demasiado teóricos, con pocos ejercicios y aún menos aplicaciones prácticas. Es algo paradójico, para un área eminentemente práctica, donde la experiencia juega un rol preponderante.

Algunos de estos problemas han podido ser resueltos gracias a la vasta difusión de los microcomputadores y a algunas de sus particularidades. La contribución del sistema UNIX también ha sido de mucha importancia, tanto por sus conceptos innovadores como por constituir un excelente banco de experimentación.

(*) Profesor de la ESAN - Apartado 1846, Lima 100.
(**) Profesor Visitante de la PUCP
Laboratoire de Génie Informatique - Institut IMAG
BP 68 38042 St. Martin d'Hères Cedex, Grenoble, Francia.

Desde hace algunos meses, ha aparecido en el mercado MINIX, una nueva versión de UNIX, con fines estrictamente educativos. Se trata de un nuevo diseño, totalmente independiente de AT&T, e implementado en un PC-IBM; el sistema es vendido por un precio bajísimo y es entregado con sus fuentes. Un sistema de este tipo representa una excelente oportunidad para los profesores de Sistemas Operativos, lo que explica la respuesta de la comunidad informática mundial: ediciones agotadas, creación de fóruns reservados a MINIX, cursos remodelados en función de MINIX, etc.

Es nuestra opinión que las Universidades del Perú y América Latina deben seguir este movimiento, tratando por supuesto de adecuarlo a nuestra realidad: es lo que tratamos de hacer en ESAN y la Universidad Católica del Perú. El objetivo de este artículo es de compartir nuestra experiencia con las personas interesadas y presentarles nuestras inquietudes al respecto.

Introducción.

La enseñanza en el área de los Sistemas Operativos siempre se ha visto confrontada a una serie de dificultades, especialmente en nuestros países, donde las Universidades sufren de una escasez notoria de recursos:

- a) Disponibilidad de máquinas con las cuales experimentar.
- b) Restricciones debidas a las características del sistema de estas máquinas (multiproceso, tiempo compartido, niveles de protección, acceso a los dispositivos hardware, etc).
- c) Falta de herramientas de experimentación.
- d) Complejidad de la interfase que define las rutinas de base de un sistema (creación de procesos, comunicación/sincro-

nización entre procesos, uso de la memoria virtual, entradas/salidad, etc).

e) Falta de homogeneidad en el diseño de las máquinas.

Estos problemas conducen en muchas ocasiones a la realización de cursos casi exclusivamente teóricos, situación paradójica para un área eminentemente práctica, donde la experiencia juega un rol preponderante.

Algunos de estos problemas han sido resueltos gracias a la vasta difusión de los micro-computadores y, en especial, de los PC-IBM y compatibles. Efectivamente, el bajo precio de éstos, aunado a las condiciones especiales de venta que benefician al sector enseñanza, han permitido equiparse a la mayoría de las Universidades e inclusive a muchos de sus alumnos. Por otro lado, la simplicidad del sistema operativo de estas máquinas (MS-DOS, CP/M u otros) ha jugado en nuestro favor; así, por ejemplo, la ausencia de mecanismos de protección y el fácil acceso a los dispositivos de la máquina han permitido diseñar trabajos de mayor envergadura (tratamiento de interrupciones, drivers de E/S) e inclusive proyectos completos (sistema tiempo real, sistema de archivos, etc). Otra consecuencia de la popularidad de estas máquinas es la facilidad con la que uno puede conseguir todo tipo de informaciones sobre ellas: descripción de los vectores de interrupción, interfase de escritura de un driver de E/S, manuales hardware, etc.

La contribución del sistema UNIX [Ritchie 74, Bourne 82] también ha sido importante, tanto por los conceptos innovadores que introdujo como por las facilidades que brinda para la realización de todo tipo de herramientas. La mayor parte de los especialistas modernos en Sistemas Operativos se han formado a través de UNIX [Bach 85], tanto por haberlo usado intensivamente como por haber construido herramientas para él o haberlo portado ha-

cia otras máquinas. Sin embargo, la complejidad de este sistema, unida a las restricciones de acceso a sus fuentes, dificultan su uso en la parte práctica de la enseñanza de Sistemas Operativos.

Desde hace algunos meses, ha aparecido en el mercado un sistema con fines estrictamente educativos, denominado MINIX [Tanenbaum 87]. Se trata de un nuevo diseño e implementación de UNIX, totalmente independiente de AT&T, que ha sido realizado por el Profesor Andrew Tanenbaum de la Universidad de Vrije (Amsterdam). Una característica importante de MINIX es que ha sido implementado en los PC-IBM, lo que le garantiza un público bastante amplio; por otro lado, el sistema es vendido por un precio irrisorio (alrededor de 80 dólares) y bajo una modalidad bastante novedosa: el cliente está autorizado a hacer copias del sistema, inclusive para otras personas. Otra característica interesante de MINIX es que el sistema es entregado con sus fuentes, lo que permite al usuario modificarlo y regenerarlo.

Es obvio que un sistema de este tipo representa una excelente oportunidad para los profesores de Sistemas Operativos. Así lo entendió la comunidad informática mundial, que ha demostrado un inmenso interés por MINIX: ediciones agotadas, creación en las principales redes mundiales de fórums reservados a MINIX, desarrollo de una gran cantidad de nuevas herramientas para él, etc. Desde entonces, muchos cursos de Sistemas Operativos han sido remodelados en función de MINIX, tanto en Estados Unidos como Europa. Nosotros estamos tratando de hacer lo propio en nuestro medio, sin perder de vista nuestra realidad.

El objetivo de este artículo es de compartir nuestra experiencia con la comunidad informática del país y de afinarla a través de las opiniones y consejos que traiga consigo su discusión. El capítulo que sigue constituye una discusión sobre la enseñanza

de los Sistemas Operativos así como una análisis de la misma en las Instituciones del país: realidad actual, aspectos que deberían ser mejor desarrollados y, finalmente, aplicaciones de estos aspectos a nuestra realidad. El tercer capítulo trata de identificar las necesidades propias de esta enseñanza tanto a nivel software como hardware, mientras que el cuarto analiza las herramientas que podrían servir a mejorarla. Finalmente, el capítulo quinto presenta de manera general el sistema MINIX e indica como pensamos hacer uso de esta herramienta para tratar de superar los problemas actuales.

Primera Parte

Enseñanza en el area de Sistemas Operativos

1. Aspectos generales

La enseñanza en el área de Sistemas Operativos cubre tópicos bastante diferentes [Deitel 83, Krakowiak 86, Tanenbaum 87] : noción de proceso, técnicas de distribución o scheduling, sincronización y comunicación entre procesos, problemas de bloqueo o deadlock, técnicas de administración de la memoria, entradas/salidas, control de dispositivos de E/S, sistemas de archivos, mecanismos de protección, seguridad, redes, etc. En la realidad, muchos de estos tópicos pueden no ser abordados o vistos de manera superficial, en función del auditorio al que se dirige el curso: informáticos, ingenieros, administradores de empresas.

Los especialistas del área están [Metzner 82, Deitel 83, Brumfield 86, ...] de acuerdo en que todo curso debe comprender los cinco grandes aspectos de un sistema: administración de procesos, entradas/salidas, administración de la memoria, siste-

mas de archivos y lenguajes de comandos. La importancia acordada a cada aspecto depende de los objetivos del curso; sin embargo, debe tratar de darse por lo menos una visión general de cada uno de ellos.

2. Enseñanza de Sistemas Operativos en el Perú

La enseñanza de los Sistemas Operativos en el Perú no sigue el esquema presentado en el párrafo anterior, limitándose a un estudio de los aspectos externos (lenguaje de comandos, explotación de un sistema) y orientándose por lo general hacia un único sistema; según tenemos entendido, son muy pocas las excepciones a esta situación.

El resultado es una visión distorsionada del área y un nivel relativamente bajo de conocimientos; las dificultades para cambiar de sistema operativo a nivel de usuario son una de las tantas consecuencias de estas deficiencias. Lamentablemente, esta figura se da tanto en las carreras técnicas como en las carreras de Ingeniería, sin tomar en cuenta los innumerables cursos ofrecidos por academias o centros de extensión de Universidades u otros organismos.

3. Proposiciones para nuestro medio

Es nuestra opinión que una enseñanza más completa en esta área redundaría en beneficio de todos. Esta enseñanza deberá incluir ciertos aspectos que consideramos indispensables:

- manejo de interrupciones, con énfasis en las oportunidades de paralelismo que ofrecen,
- realización de controladores o drivers de entrada/salida simples,
- núcleo de un sistema, haciendo hincapié en la noción de pro-

ceso y en las relaciones que se pueden establecer entre dos procesos,

- técnicas de administración de la memoria, tratando de dar una mayor importancia a aquellas que pueden ser útiles en nuestro medio (todas las relativas a zonas contiguas de memoria),
- nociones principales asociadas a un sistema de archivos: organización, archivos compartidos, protección entre usuarios, seguridad, distribución en una red, etc,
- uso de rutinas sistema, mostrando un ejemplo de interfase para la programación sistema,
- lenguaje de comandos en un sentido más amplio, apoyándose en un lenguaje moderno, en el que se pueda realmente programar.

Todos estos aspectos deberán ser enfocados con el mayor sentido práctico y a través de ejemplos de diferentes sistemas, de manera a dar una visión más general.

Son numerosas las aplicaciones directas, en nuestro medio, de un curso de este estilo:

- mejor explotación de los recursos de una máquina y, por supuesto, de su sistema,
- mejor explotación de un gran número de productos (utilitarios, bases de datos, aplicaciones finales),
- formación de verdaderos ingenieros sistema,
- desarrollo de software de base más apropiado a una empresa o, en general, a nuestra realidad,
- realización de aplicaciones tiempo real (control industrial, burótica, juegos),
- desarrollo de drivers para tarjetas o dispositivos especiales, con aplicaciones en Ingeniería (industrial, mecánica, eléctrica, etc),

- realización de aplicaciones de tipo teleproceso,
- mejor comprensión y, por ende, mejor explotación de redes,
- ...

En el capítulo siguiente, analizamos las necesidades propias a este tipo de curso, para discutir luego su implementación.

Segunda Parte

Necesidades para la enseñanza de Sistemas Operativos

La enseñanza de los Sistemas Operativos se ha caracterizado en muchos casos por su orientación demasiado teórica y su escasa o inexistente parte práctica. Las dificultades enumeradas en el capítulo I son directamente responsables de esta situación, que se ve agravada por la falta de experiencia de muchos profesores.

Consideramos que se trata de un problema fundamental, que debe ser resuelto a través de la introducción de un conjunto coherente de ejercicios y trabajos prácticos; ésto sólo podrá ser hecho si existe la infraestructura apropiada, tanto a nivel hardware como software.

Un fenómeno de gran utilidad para nuestros propósitos, es el nivel de equipamiento en micro-computadores al que han llegado las Universidades en estos últimos años. Efectivamente, los micro-computadores constituyen las mejores herramientas a nivel hardware para la enseñanza de los Sistemas Operativos, pues:

- se trata de máquinas mucho más simples,
- el usuario que los usa no interfiere con los demás, como en

- el caso de máquinas que usan sistemas multi-usuario,
- el acceso a las informaciones sobre la máquina o su sistema no está restringido (interfase de llamadas sistema, vectores de interrupción, estructura de los drivers E/S, descripción de los dispositivos de E/S, etc.),
- el sistema operativo nativo es abierto y permite modificar o agregar ciertas características,
- el sistema operativo puede ser reemplazado por otro sistema.

Por supuesto, estas características no se dan en su totalidad en todo micro-computador; así, por ejemplo, el PC-IBM responde mucho mejor a ellas que el Macintosh.

En cambio, en el aspecto software, es mucho más difícil contar con herramientas apropiadas y, sobre todo, a bajo costo. El capítulo que sigue analiza las dificultades en esta área y las soluciones usadas en otros lugares.

Tercera Parte

En búsqueda de una herramienta apropiada

1. La influencia de UNIX

La carencia de herramientas para la enseñanza de los Sistemas Operativos de estos últimos años llevó a numerosas Universidades a definir sus propias políticas de desarrollo, con el objetivo de dotarse de una infraestructura apropiada en esta área.

La mayor parte de los proyectos emprendidos fueron sumamente influenciados por los conceptos introducidos por el sistema UNIX, cuando no se basaron directamente en él; la contribución

de UNIX ha sido en este sentido muy importante en él; la contribución de UNIX ha sido en este sentido muy importante. Cabe destacar que la mayoría de los especialistas modernos en Sistemas Operativos se han formado con este sistema, sea portándolo hacia otras máquinas o desarrollando nuevas herramientas para él (paginación, núcleo multi-procesador, nuevos drivers, interfaces más amigables, etc); es toda una generación de investigadores, ingenieros y estudiantes que se han servido de UNIX como base de experimentación.

2. Restricciones en el uso de UNIX. Nuevas herramientas

Un factor preponderante en la importancia adquirida por UNIX fue la amplia disponibilidad de sus fuentes, cedidos bajo licencia por AT&T tanto a las Universidades y Centros de Investigación como a las empresas privadas. Lamentablemente, esta misma importancia convirtió a UNIX en un producto comercial estratégico, lo que llevó a AT&T a revisar su política y a restringir el acceso a los fuentes del sistema. Es así que a partir de la V7 de UNIX, las Universidades sólo podían emplear estos fuentes con fines de investigación y no en sus cursos de Sistemas Operativos. Es en ese contexto que aparecen numerosos proyectos, que tratan de llenar el vacío creado por las restricciones de AT&T: **Tunis** [Holt 83] en la Univ. de Toronto, **Xinu** [Comer 84] proyecto conjunto de numerosas Universidades, **MINIX** [Tanenbaum 87] en la Univ. de Vrije (Amsterdam), etc.

Dentro de estos proyectos, destaca nítidamente MINIX. Se trata de un nuevo sistema, compatible con la V7 de UNIX pero completamente independiente de AT&T; efectivamente, el sistema ha sido diseñado y realizado desde cero, sin usar ninguna línea de código de AT&T, tanto en lo que concierne al sistema en sí mismo como a sus utilitarios. MINIX no está por tanto, sometido a las mismas restricciones de acceso a los fuentes que UNIX.

3. Política de distribución de MINIX

La política de distribución bajo la cual ha aparecido MINIX merece un comentario aparte. Efectivamente, el Profesor Andrew S. Tanenbaum publicó a comienzos de 1987 un nuevo libro sobre Sistemas Operativos, en el cual se sirve de un mini-UNIX (de donde deriva el nombre de MINIX) para explicar todos los conceptos teóricos desarrollados en su libro; al mismo tiempo, hizo circular en la red internacional USENET un mensaje que anunciaba la publicación de su libro, así como una breve descripción de MINIX y de sus condiciones de venta. Este mensaje despertó una enorme expectativa en la comunidad informática, reflejada por el agotamiento de tres ediciones sucesivas del libro y por la creación de numerosos fórums de discusión sobre MINIX.

Cabe indicar que el libro del Profesor Tanenbaum hace uso intensivo de los fuentes del sistema, publicados en anexo del libro; de esta manera, los lectores pueden ver como ha sido escrito el sistema, cuáles son los algoritmos usados en cada aspecto, las particularidades de la implementación, etc. Son estos fuentes que permitieron las primeras discusiones en los fórums reservados a MINIX.

Para permitir una real experimentación con el sistema, la publicación del libro fue seguida dos meses después por la del software, con características realmente innovadoras:

- la edición incluye el ejecutable y los fuentes del sistema,
- el precio de venta es irrisorio (un poco menos de 80 dólares), considerando el costo de las diskettes (8 en total), el empaque y la comercialización,
- la publicación y comercialización es hecha por una de las grandes editoriales científicas (Prentice-Hall),
- la licencia del uso del software permite la copia de los dis-

kettes tanto para uso propio como para el de otras personas (dentro de límites "razonables")

- la misma licencia permite que los estudiantes de un curso puedan duplicar y hacer uso del software para fines educativos, sin necesidad de adquirir varios ejemplares,
- finalmente, esta misma licencia permite la modificación de los fuentes y la regeneración del sistema, con la única restricción de hacer llegar al autor los cambios que puedan ser benéficos al resto de usuarios.

Por supuesto, la expectativa despertada fue aún mayor, después de la edición del software; para muestra un botón: entre los meses de abril y julio de 1987, fueron registrados alrededor de 1200 artículos en el fórum MINIX de USENET, convirtiéndolo en uno de los principales de la red.

El interés por MINIX se ha manifestado concretamente a través de numerosas proposiciones de mejora del sistema y de la implementación en diversos lugares de cursos sobre Sistemas Operativos basados en MINIX. Las mejores propuestas (purga de errores, cambios en ciertos módulos, nuevos utilitarios) han inclusive llevado a la edición de una nueva revisión (release) del software que ha circulado gratuitamente.

Cuarta Parte

Uso de MINIX para la enseñanza de Sistemas Operativos

Una conclusión que podríamos sacar del capítulo anterior es que MINIX (libro + software) constituye una herramienta excepcional para la enseñanza de los Sistemas Operativos. El libros en sí cubre la mayor parte de los aspectos propuestos en el párrafo II.3;

por otro lado, la posibilidad de usar los fuentes del sistema permite planificar una parte práctica que dé al alumno una sólida experiencia. En el párrafo siguiente, presentamos un cuadro general de MINIX, para luego proponer un conjunto de ejercicios y proyectos que nos parecen apropiados.

1. Presentación general de MINIX

MINIX está basado en la versión 7 de UNIX, considerada por todos como la versión más simple y elegante de este sistema, aunque no la más poderosa. Las llamadas sistema de MINIX son exactamente las mismas de la V7; por otro lado, el language de comandos es funcionalmente idéntico al shell UNIX e incluye una gran cantidad de utilitarios similares a los de la V7. Desde su aparición, han sido desarrollados un nuevo shell y muchos otros utilitarios, que lo aproximan aún más a UNIX.

MINIX ha sido realizado en C (muy pocas rutinas son escritas en Assembler) e implementado inicialmente sólo para los PC-IBM. Sin embargo, varios transportes hacia otras máquinas están actualmente en curso: Atari, Macintosh, Amiga, etc; la idea de base es implementarlo en los micro-computadores más populares, para aumentar la "audiencia" del sistema.

Dado que MINIX ha sido hecho con fines educativos, el autor ha sido sumamente cuidadoso en la realización. La estructura del sistema es bastante modular, para facilitar su comprensión y su modificación; así mismo, han sido incluidos una gran cantidad de comentarios (prácticamente un cuarto del código), que explican el programa.

El sistema está organizado en cuatro grandes componentes: administración de procesos, entradas/salidas, administración de la memoria y sistema de archivos. Estos componentes han

sido realizados como módulos independientes que se comunican a través de mensajes; todo el sistema está basado en esta técnica (message passing), lo que le confiere una gran flexibilidad, aunque le reste eficiencia. Cada uno de los componentes es representado durante la ejecución del sistema como un proceso independiente, con su propio espacio de direccionamiento e informaciones de estado; este esquema de funcionamiento obliga a definir una interfase clara entre los procesos y resulta en una mejor protección entre ellos.

A la generación del sistema, cuatro programas ejecutables independientes son creados: el núcleo o kernel, el administrador de la memoria, el sistema de archivos e "init". Este último programa se encarga de escrutar las líneas de comunicación para ver si algún usuario desea conectarse y entabla el diálogo correspondiente al "login", creando los procesos necesarios. Los cuatro programas pueden ser compilados separadamente para formar un nuevo sistema, lo que limita las recompilaciones.

El kernel incluye los drivers de entrada/salida, pero cada uno de éstos es creado como un proceso independiente. Cabe indicar que todos los procesos sistema compiten con los procesos usuarios por la posesión del procesador; estos procesos pueden por consiguiente ser bloqueados, si alguno de sus mensajes no tiene respuesta inmediata. De todas maneras, una escala de prioridades es necesaria, para un mejor funcionamiento del sistema; es por esta razón que los procesos han sido estructurados en cuatro niveles:

- 1) Manejo de los procesos y de los mensajes de comunicación. Se trata del nivel más bajo, encargado de la administración de los procesos y, por consiguiente, de la distribución del procesador y de la comunicación entre procesos. La técnica de distribución es bastante simple y privilegia los procesos en

función de sus niveles.

- 2) Drivers de entrada/salida. Estos procesos forman parte del Kernel, de manera a poder acceder a los dispositivos de E/S y a las intrucciones que permiten manipularlos. Los procesos del kernel son llamados tareas, para distinguirlos de todo el resto. Las interrupciones son capturadas por el nivel 1, que determina el driver encargado del tratamiento correspondiente y le envía un mensaje, pidiéndole que se ocupe de ella.
- 3) Administrador de la memoria y sistema de archivos. Estos dos procesos cumplen el rol de servers, que atienden los pedidos de los usuarios (sus "clientes"). Efectivamente, cuando un usuario necesita memoria o usar algún archivo, debe efectuar una llamada sistema que es transformada por la rutina correspondiente en un mensaje al server implicado; la rutina es bloqueada en ese instante (y, por ende, el usuario), hasta que su pedido haya sido satisfecho y el mensaje respondido con los resultados de la operación. El tratamiento del pedido puede necesitar la participación de varios procesos y la realización de diferentes acciones, de manera transparente al usuario.
- 4) Procesos de los usuarios, que incluyen tanto los procesos iniciales (creados en respuesta a un login) como aquellos que son creados directamente por el usuario, a través de la primitiva "fork". Estos procesos constituyen un árbol, cuya raíz es el proceso "init".

2. Uso de MINIX como ejemplo de sistema

MINIX es estrictamente similar a UNIX V7 a nivel del usuario. Por esta razón, MINIX constituye un excelente ejemplo de sistema, pues incorpora todas las innovaciones de UNIX: un verdadero lenguaje de comandos, las redirecciones, los pipes, el sistema de protección de archivos en anillos, un sistema de archivos arbo-

rescente, etc.

Por otro lado, los fuentes del sistema y el libro que lo describe, convierten a MINIX en un instrumento pedagógico de primer orden, que puede ser descuartizado a la manera de una disección en biología.

Finalmente, debemos agregar una tremenda ventaja de MINIX: ha sido desarrollado para los PC-IBM, los micro-computadores de mayor difusión en nuestro medio, e inclusive funciona en configuraciones reducidas de éstos (256K de memoria y sin disco duro).

3. Uso de MINIX como herramienta de experimentación

Aparte de su uso como ejemplo de sistema operativo, MINIX puede servir como herramienta de experimentación, sobre la cual se ejerciten los alumnos de un curso: es en este campo que MINIX ofrece sus mejores servicios.

Es difícil proponer un conjunto de trabajos prácticos que se adecúen a todas las circunstancias: objetivos del curso, nivel de los alumnos, orientación de la carrera, etc. El propósito de este párrafo es más bien de contribuir a formar una biblioteca de ejercicios y proyectos, que podría ayudar a otros profesores a planear un curso.

Los trabajos que proponemos se descomponen en dos categorías: trabajos cortos o asignaciones y trabajos de mayor envargadura o proyectos. Entre los primeros, proponemos los siguientes:

- a) Uso del shell (lenguaje de comandos de MINIX) para programar utilitarios destinados a los operadores del sistema: realizar un backup del sistema de archivos, borrar los archivos

de tipo backup creados por ciertas aplicaciones, implementar un comando find (buscar un archivo y someterlo a un tratamiento dado).

- b) Modificaciones triviales del sistema. El objetivo de este trabajo sería de mostrar a los alumnos como se genera un sistema operativo, después de haber sido modificado. Las modificaciones en sí mismas no tienen mucha importancia: la corrección de ciertos errores de MINIX, catalogados por su autor pero aún no corregidos en la versión actual, constituye un buen ejercicio.
- c) Familiarización con las llamadas sistema de MINIX, a través de un programa C que cree procesos y los haga comunicar (primitivas fork, exec, pipe).
- d) Pequeños programas que hagan uso de dispositivos internos de la máquina. La actualización de la fecha y la hora, a partir del llamado "reloj tiempo real", es un buen ejemplo pero sólo podría ser efectuado en ciertas máquinas (PCs con tarjeta multi-función o PC-AT). Otra posibilidad es la realización de un pequeño driver para manipular el sintetizador del PC.
- e) Estudio del kernel de MINIX. Tres trabajos interesantes pueden ser realizados en esta rúbrica:
 - Trazar los procesos que se ocupan de las interrupciones, las llamadas sistema y del envío/recepción de mensajes, mostrando el interés del multi-proceso.
 - Calcular el tiempo ocioso de la máquina, es decir, aquel durante el cual ningún proceso está activo.
 - Cambiar la política de distribución del procesador, dividiendo los procesos de los usuarios en dos categorías: interactivos y de cálculo; estos últimos deberán tener una mayor prioridad. El kernel deberá determinar la categoría de cada proceso dinámicamente.
- f) Mejorar la técnica de administración de la memoria usada por MINIX. Con la estrategia actual, la creación de un

proceso puede ser rechazada si no hay suficiente espacio contiguo; sin embargo, la muerte de otros procesos ha podido liberar varios espacios libres que sumados podrían ser suficientes para lograr la creación del proceso. La nueva estrategia consistiría en compactar la memoria, copiando todos los procesos activos lo más bajo posible en la memoria, de manera a concentrar todo el espacio libre al final de la memoria.

- g) Implementación de nuevos comandos de manipulación de archivos: tail (impresión de la última parte de un archivo) cpdir (copiar todo un directorio), whereis (donde se encuentran las fuentes y el ejecutable de un programa). Este trabajo permitiría a los alumnos adquirir un buen conocimiento sobre el sistema de archivos de MINIX.

El tiempo que tomaría cada una de estas asignaciones puede variar bastante, en función de las condiciones de trabajo: disponibilidad de máquinas, nivel de los alumnos, apoyo del profesor. Sin embargo, un plazo de tres semanas para cada uno, nos parece bastante razonable, pues se trata de trabajos relativamente cortos.

En cambio, los proyectos son trabajos mucho más importantes, que necesitan mucho más tiempo (2 a 3 meses) y sólo deben ser propuestos en ciertos casos: curso de larga duración, segundo curso en Sistemas Operativos, curso para especialistas, proyecto que evalúa todo o parte importante de un curso, etc. Tres proyectos nos parecen de un gran interés:

- a) Implementación de un spool de impresión, encargado de copiar los archivos que se quieren imprimir en un directorio especial y de imprimirlos tan pronto como sea posible (impresora en línea y desocupada). El usuario podrá mientras tanto continuar normalmente su trabajo.

- b) Realización de un driver RS-232, capaz de controlar un puerto serie del PC (COM1). Con este driver, se podrá conectar otros terminales al Pc y usarlo como un sistema multi-usuario.
- c) Uso de la técnica de swapping para administrar la memoria. Este proyecto permitiría una mayor flexibilidad en el uso de MINIX, principalmente si se cuenta con un sistema multi-usuario (proyecto b).

Cabe indicar que estos proyectos implican un buen nivel en el área y que están más bien destinados a futuros especialistas en Sistemas Operativos o futuros profesores del área. Por otro lado, las máquinas de desarrollo deberán contar con un disco duro.

4. Problemas prácticos para el uso del MINIX

Existe todo un banco de datos sobre correcciones de errores, mejoras y diversos "trucos" de MINIX, que han circulado a través de USENET. La aplicación de este banco de datos constituye un excelente medio para conocer los problemas prácticos ligados a MINIX: problemas de instalación, generación de un nuevo sistema, desviaciones de ciertos comandos, etc. Se trata de una tarea de mucha utilidad, que permite adquirir una sólida experiencia en MINIX; pensamos que es una etapa indispensable para quien desee usar MINIX como herramienta de enseñanza.

Tanto la Universidad Católica como ESAN cuentan con este banco de datos y han podido explotarlo. Nuestra experiencia puede ser útil a las personas interesadas.

Conclusiones.

La enseñanza de los Sistemas Operativos siempre se ha visto confrontada a diversos problemas que no le han permitido cumplir

cabalmente su rol. En nuestro país, estos problemas son más agudos, producto de una visión distorsionada del área.

Nuestro artículo analiza esta situación y trata de encontrar una vía hacia la solución de estos problemas. Nuestras proposiciones cubren tanto el aspecto teórico como práctico del área de los Sistema Operativos y están basadas en el uso de una nueva herramienta: MINIX.

En la actualidad, estamos tratando de poner en práctica nuestras proposiciones en dos contextos totalmente diferentes: Magister de Informática de la Universidad Católica y PADE de Informática de ESAN. Esperamos que estas experiencias nos permitan afinar y completar nuestras proposiciones, para beneficio de la enseñanza de la Informática en nuestro país.

Por otro lado, consideramos que este congreso nos brinda una excelente oportunidad para transmitir nuestra experiencia y discutir nuestras proposiciones con las personas interesadas en esta área.

Bibliografía.

[Bach 85]

Bach, "The Design of the UNIX Operating System", Prentice-Hall, 1985.

[Bourne 82]

Bourne S, "The Unix System", Addison Wesley, 1982.

[Brumfield 86]

Brumfield J, "A guide to Operating Systems Literature", OSR-ACM 20-2, Abril 86, p. 38-42.

[Comer 84]

Comer D, "Operating System Design: The Xinu approach", Prentice-Hall, 1984.

[Deitel 83]

Deitel H, "An introduction to Operating Systems", Addison Wesley, 1983.

[Holt 83]

Holt R, "Concurrent Euclid, the UNIX System and TUNIS", Addison Wesley, 1983.

[Krakowiak 86]

Krakowiak S, "Systèmes d'Exploitation", Dunod, 1986.

[Metzner 82]

Metzner J, "Structuring Operating Systems Literature for the Graduate Course", OSR-ACM 16-4, Octobre 1982, p. 10-25.

[Ritchie 74]

Ritchie D, Thompson K, "The UNIX Time-Sharing System", Communications of the ACM 17-7, Julio 1974, p. 365-375.

[Tanenbaum 87]

Tanenbaum A, "Operating Systems: Design and Implementation", Prentice-Hall International, 1987.