

Uso de simuladores y herramientas Web para la enseñanza de Sistemas Operativos

Félix Buendía, Juan-Carlos Cano, Julio Sahuquillo

Departamento de Informática de Sistemas y Computadores

Escuela Técnica Superior de Informática Aplicada

Universidad Politécnica de Valencia

e-mail: {fbuendia, jucano, jsahuqui}@disca.upv.es

Resumen

La enseñanza de las materias relacionadas con los Sistemas Operativos se organiza en varias asignaturas que cubren un conjunto de contenidos esenciales en cualquier currículo universitario de informática. Las diferentes asignaturas impartidas incluyen temas con una fuerte componente teórica y conceptual. Aunque dichos conceptos son ampliamente tratados en numerosos libros de texto, no resulta sencillo encontrar entornos adecuados al nivel de conocimientos del alumno, que permitan realizar actividades donde poner en práctica los conceptos introducidos de forma teórica. En este trabajo se describen dos herramientas que permiten la puesta en práctica de dichos conceptos a partir del uso de aplicaciones Web. La primera representa un simulador utilizado para mostrar el funcionamiento de los principales aspectos relacionados con la gestión de memoria en un sistema operativo, de forma que el alumno pueda interactuar con el simulador, introduciendo ejemplos de carga y configurando parámetros del sistema. La segunda herramienta permite el acceso remoto a funciones de un sistema operativo real a través de un interfaz Web. La experiencia docente con ambas herramientas corrobora su utilidad tanto en aspectos relacionados con la comprensión del alumno como en la evaluación por parte del profesor.

1. Introducción

Las asignaturas relacionadas con los Sistemas Operativos son un elemento fundamental en cualquier currículo universitario de informática. Desde las propuestas de ACM/IEEE de 1991 [1] hasta las más recientes del CC 2001 [2], existe un

consenso sobre la importancia de esta temática. Las directrices del MEC establecen una serie de descriptores que hacen referencia a aspectos tales como la gestión de procesos y ficheros así como los recursos de memoria y entrada/salida. Se trata de aspectos con una fuerte componente teórica y que deben encontrarse apoyados con actividades prácticas para facilitar su comprensión. Sin embargo, no resulta sencillo encontrar entornos de trabajo donde realizar este tipo de actividades que permitan al alumno fijar con la práctica los conceptos tratados de forma teórica. Por otro lado, las actividades prácticas se han realizado tradicionalmente sobre los sistemas operativos reales donde el acceso a los mecanismos internos puede resultar excesivamente complejo.

En algunos casos, se dispone de sistemas operativos reducidos como Minix [3] y Nachos [4], que aunque presentan una amplia funcionalidad, para asignaturas troncales básicas tienen como principal inconveniente su orientación hacia aspectos de diseño e implementación.

Otras herramientas como RCOS [5] o SOsim [6] simulan el funcionamiento de aspectos de un sistema operativo. Sin embargo, la propuesta de RCOS se centra demasiado en los aspectos visuales y no profundiza en la aplicación de aspectos teóricos. Magee y Kramer [7] presentan una herramienta que, de manera similar a la que se presenta en esta ponencia, se centra en la demostración visual de aspectos relacionados con los conceptos de concurrencia.

El uso de simuladores puede resultar útil para fijar los conceptos teóricos mediante actividades prácticas en asignaturas básicas. Sin embargo, su utilización no significa que haya que renunciar al trabajo con sistemas operativos reales. Una de las capacidades prácticas que recomienda el CC 2001

es la de “operar con equipamiento y software de sistema de manera eficiente”. Los sistemas operativos suelen incluir asistentes que muestran al usuario su funcionamiento o guían a éste en su gestión. Sin embargo, existen pocas herramientas que permitan el trabajo práctico con sistemas operativos reales utilizando un enfoque didáctico. Las herramientas que se describen en este trabajo representan dos visiones de actividades prácticas relacionadas con el manejo de sistemas operativos y que pueden complementarse para facilitar una docencia más eficaz en esta disciplina.

El resto del trabajo se organiza como sigue. En la Sección 2, se presenta el simulador de la gestión de memoria de un sistema operativo accesible vía Web. La Sección 3 describe herramientas basadas en aplicaciones Web que permiten el acceso a funciones del sistema operativo de una forma didáctica. La Sección 4 refleja la valoración sobre el uso docente de estas herramientas y su papel en las nuevas metodologías que se avocan con la integración en el espacio europeo de educación superior (EEES). Finalmente, la Sección 5 presenta las conclusiones del trabajo.

2. Simulador de Gestión de Memoria

La herramienta de simulación ha sido desarrollada en lenguaje Java. Una primera versión de la misma se desarrolló en el contexto de un proyecto final de carrera dentro de la Escuela de Informática de la Universidad Politécnica de Valencia [8]. Nuevas versiones con interfaz mejorada se han ido incorporando como parte del trabajo realizado en el ámbito de un proyecto europeo de educación a distancia [9].

2.1. Introducción al simulador

El Simulador de gestión de memoria tiene por objeto mostrar el funcionamiento de algunos aspectos de la gestión de memoria que intervienen en la ejecución de uno o más procesos en un entorno de multiprogramación. Concretamente, la herramienta describe la ejecución de un conjunto de procesos mediante el algoritmo de planificación *round-robin* y cómo estos procesos generan durante su ejecución accesos a direcciones lógicas que serán traducidas a direcciones en memoria física. El esquema de

traducción de direcciones se basa en una técnica de paginación combinada con algoritmos de reemplazo, lo que se conoce globalmente como memoria virtual mediante paginación bajo demanda.

2.2. Inicio de una sesión

El primer paso en la ejecución de una simulación de gestión de memoria consiste en cargar el fichero de simulación (menú Archivo). Para describir un determinado programa, se ofrece un fichero denominado “Simulacion.inf” que describe las características de los diferentes procesos. Este fichero contiene una secuencia de direcciones lógicas agrupadas por cada proceso que se incluye en el sistema. Para ello se utiliza una cabecera con tres campos: el primero hace referencia al identificador del proceso, el segundo el tamaño asignado a dicho proceso y el tercero a la prioridad, que identificará la cola a la que pertenece el proceso. La Figura 1 muestra un ejemplo.

```
Proceso1 50000 0
5000
0
100
0
456
4000
1050
end
0...
```

Figura 1. Ejemplo de fichero Simulacion.inf.

En dicho ejemplo la primera línea del fichero contiene la siguiente declaración:

```
Proceso1 50000 0
```

que define el Proceso1, con un tamaño de 50000 bytes y una prioridad 0. A partir de esta línea aparecen valores numéricos separados en diferentes líneas, cada uno de los cuales representa una dirección lógica. Las direcciones lógicas contenidas en un mismo proceso finalizan cuando aparece la etiqueta “end”. A continuación, y antes de empezar la especificación de un nuevo proceso, aparece un valor numérico que especifica el desplazamiento temporal en la activación del siguiente proceso respecto al anterior. En este caso, se utiliza un valor 0 que indica que el Proceso1 y el siguiente (Proceso2) se activan al mismo tiempo. Un valor de 2 significaría que

deben transcurrir dos instantes de tiempo antes de que se active el Proceso2.

2.3. Configuración del sistema

Una vez que se ha cargado el fichero de simulación, se debe proceder a la configuración del sistema mediante la opción de dicho nombre del menú Archivo. La Figura 2 muestra un ejemplo. Se puede observar cómo esta opción permite asignar diferentes atributos del sistema tales como el tamaño de la memoria, el tamaño de la página, los algoritmos de asignación y reemplazo del sistema.

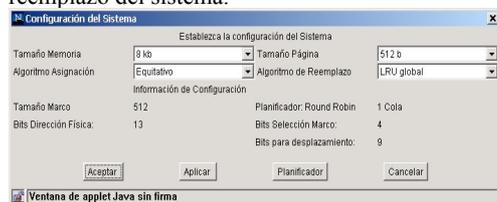


Figura 2. Ventana de configuración del sistema.

2.4. Comienzo de la simulación

Una vez cargado el fichero de simulación y realizada la configuración del sistema, para simular el comportamiento se elige la opción “Comenzar Simulación” dentro del menú Archivo.

Una vez iniciada la simulación, se puede observar la evolución de la misma a través de diferentes zonas de la pantalla mostradas en la Figura 3. Existen dos formas de controlar la simulación, las cuales se pueden seleccionar desde el menú de Simulación:

- Modo paso a paso: permite visualizar los resultados de cada paso (unidad temporal) en que se realiza cada acceso a memoria o la activación de un nuevo proceso.
- Modo automático: la ejecución de los procesos y, por tanto, el acceso a direcciones de memoria se realiza de forma automática sin intervención del usuario.

La parte superior izquierda de la pantalla muestra una representación gráfica de los procesos activos en cada cola. En el ejemplo se utiliza una única cola (Cola 0) que contiene los tres procesos definidos. Además, cada vez que se

accede a la dirección lógica de un proceso, éste se pondrá al inicio de la cola y cambiará el color del símbolo que lo representa, para indicar que es el proceso en ejecución.

La parte superior derecha de la pantalla informa del estado del proceso en ejecución. Para ello, se incluye el contador de direcciones lógicas, el número de accesos realizados hasta el momento actual y el valor de la tasa de fallos de página.

La parte inferior izquierda representa la gestión de páginas e incluye información relativa al algoritmo de reemplazo seleccionado, el tipo de asignación de marcos si la hubiera y la lista de páginas utilizadas por cada proceso y sometidas al correspondiente algoritmo de reemplazo.

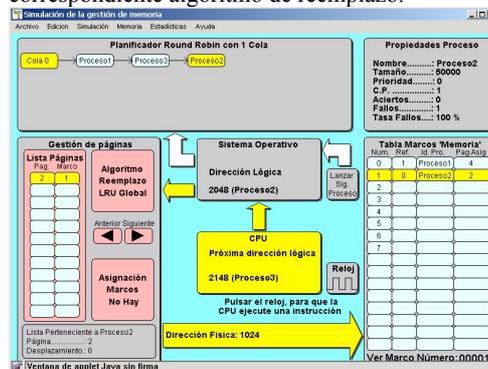


Figura 3. Ejemplo de ejecución del simulador.

La parte inferior derecha de la pantalla representa la Tabla de Marcos de la memoria física. En ella se muestra la siguiente información:

- El número de marco: su valor va desde 0 hasta N, siendo N+1 el número de marcos disponible.
- El campo referencia define la antigüedad de las páginas que se almacenan en memoria.
- El identificador de proceso: indica el proceso que accede a la página.
- Página asignada: se trata de la página almacenada en el marco referido.

Por último, la parte central de la pantalla define la relación entre las estructuras descritas previamente. Para ello, se indica en cada momento qué proceso ha sido elegido por parte del planificador para su ejecución y la dirección lógica implicada en este paso. También se encuentran en esta parte de la pantalla tres etiquetas que contienen la siguiente información:

- Sistema operativo: indica la dirección lógica que actualmente está siendo traducida.
- CPU: representa la próxima dirección a traducir.
- Dirección física: contiene la dirección física resultado de la traducción actual.

2.5. Resultados de la simulación

Mediante la opción “Ver Ventana de Direcciones Físicas” del menú Edición, se obtiene una traza de la simulación. Esta traza muestra un listado de direcciones físicas como consecuencia del proceso de traducción.

Tras finalizar la simulación y descargar los procesos de la memoria, se pueden obtener estadísticas de la ejecución de la simulación. La Figura 4 muestra un ejemplo de dichos resultados, los cuales se pueden obtener utilizando las opciones “Tasa de Fallos” y “Rendimiento” dentro del menú “Estadísticas”.

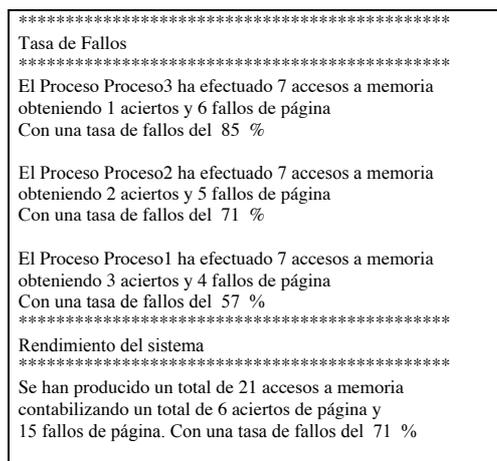


Figura 4. Estadísticas de la simulación.

3. Herramienta de acceso al sistema operativo

Esta sección describe una herramienta basada en aplicaciones Web que permiten el acceso a los servicios de un sistema operativo de una manera supervisada y didáctica. Los sistemas operativos raramente utilizan una interfaz Web como medio de acceso remoto y se emplean otros mecanismos

como conexiones *telnet* o *rsh* en entornos UNIX o servicios de Terminal Server en Windows. Existen productos como HTSH [10] o PHPShell [11] que permiten la ejecución de programas shell a través de la Web, o WSH [12] dirigido a ejecutar programas en una máquina Windows remota pero sin una utilidad didáctica específica.

3.1. Antecedentes: RPEWA

La herramienta descrita se basa en una aplicación denominada RPEWA (Remote Program Execution Web Application) cuyo principal objetivo consistía en permitir a estudiantes de sistemas operativos la carga y comprobación de programas escritos en lenguaje *C* o *shellscript*. Esta aplicación se desarrolló como un proyecto final de carrera y su esquema se muestra en la Figura 5.

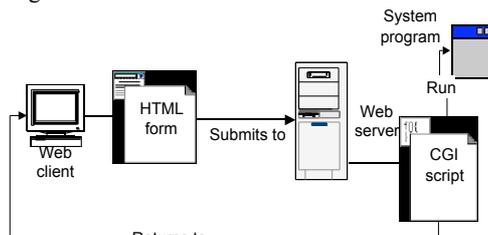


Figura 5. Esquema de RPEWA.

RPEWA funciona como sigue. En primer lugar, el cliente Web (p.e. un navegador) accede a una dirección (URL) que devuelve un formulario HTML que permite al alumno introducir el código del programa y sus parámetros. Una vez introducida esta información en el formulario se envía al servidor Web que a su vez la procesa mediante un *script* CGI programado en lenguaje Perl. Finalmente dicho *script* ejecuta el programa transferido y devuelve su resultado al cliente Web.

Esta aplicación funciona de forma autónoma, lo que le proporciona una gran flexibilidad a la hora de poder introducir ejemplos de código y realizar la comprobación de su funcionamiento. El hecho de poder utilizarse desde un navegador Web facilita su acceso universal y la posibilidad de utilizar servicios del sistema operativo no disponibles en la máquina cliente. Además proporciona al alumno una representación de las salidas estándar y de error de los programas ejecutados, así como la opción de guardar dichos

programas y recuperarlos en futuras sesiones de trabajo. Esta herramienta es un claro ejemplo del uso de aplicaciones Web para facilitar el acceso a servicios de un sistema operativo y, por tanto, de contribuir a su aprendizaje. Sin embargo, se detectaron ciertas carencias en su concepción que dieron lugar a una nueva herramienta que se describe en la siguiente sección.

3.2. RPEWA 2

Una de las carencias detectadas consistía en la falta de relación entre las actividades realizadas mediante RPEWA y los contenidos teóricos donde se podían enmarcar dichas actividades. Otra de las cuestiones pendientes en RPEWA era la ausencia de un entorno desde el que se pudiese controlar o supervisar el uso docente de esta herramienta.

Para solucionar estas carencias se integraron los mecanismos de RPEWA en un entorno de campus virtual denominado *Xedu* [14] dando como resultado una nueva versión denominada RPEWA2. Dicho entorno permite registrar y controlar el acceso de los diferentes usuarios a los contenidos y actividades de una determinada asignatura. Además se incorporó el uso de una entidad denominada *Unidad de Aprendizaje* (Learning Unit) [15] que permitía integrar los contenidos teóricos y las actividades prácticas para su mejor aprovechamiento.

Una *Unidad de Aprendizaje* consta de una serie de elementos para representar recursos didácticos expresados mediante un lenguaje de marcado denominado LMML (Learning Material Mark-up Language) [16]. Dichos elementos estructuran la información de tipo teórico a través de etiquetas como *paragraph* y los ejercicios o actividades prácticas (etiqueta *exercise*)

La Figura 6 muestra parte de un documento LMML asociado a una *Unidad de Aprendizaje* que se utiliza para la enseñanza de Sistemas Operativos en el apartado de gestión de procesos. En este caso se trata de explicar la finalidad de un intérprete de órdenes en entornos UNIX, y cómo éste se encarga de ejecutar órdenes simples como *ls* o *who*. Con el fin de reforzar la adquisición de estos conceptos teóricos se propone la realización de un ejercicio (elemento *exercise*) en el que el alumno tiene que elaborar un programa en lenguaje C que se encargue de interpretar las ordenes mencionadas.

```

<lmml>
  <collection title="Intérprete de órdenes sencillo (0)"
  difficulty="low">
    <motivation>
      <LMMLtext>Se trata de diseñar y
      codificar, en lenguaje C y sobre un sistema operativo
      Linux, un programa que actúe como intérprete de
      órdenes sencillo</LMMLtext>
    </motivation>
    <section title="Ejecución de órdenes
    simples">
      <paragraph>
        <LMMLtext>En este caso, se
        pretende ejecutar órdenes simples como
        <emphasized>ls</emphasized> o <emphasized>
        who</emphasized></LMMLtext>
      </paragraph>
      <exercise>
        <LMMLtext>Generar una versión de
        ush interactiva con el nombre ush0.c y cuya sintaxis de
        llamada sea: </LMMLtext>
        <code><![CDATA[ush
        <orden>]]</code>
        <LMMLcode type="input" style="file"
        uri="ush0.c"/>
      </exercise>
      <remark>
        <LMMLtext>Se trata de utilizar la
        llamada a sistema <emphasized>exec</emphasized>
        tal como se explica en</LMMLtext>
        <LMMLcode type="output"
        uri="http://sop.upv.es/so2/pdf/UT1.pdf" title="Tema 1
        (SO2)"/>
      </remark>...
  </section>
</collection>
  
```

Figura 6. Ejemplo de Unidad de Aprendizaje.

La Figura 7 muestra un esquema de la aplicación RPEWA2 que se encarga de procesar los documentos asociados a las *Unidades de Aprendizaje* dentro del entorno *Xedu*. El funcionamiento también se produce desde un cliente Web que accede a una URL implementada mediante un documento XML situado en un servidor Web bajo *Xedu*.

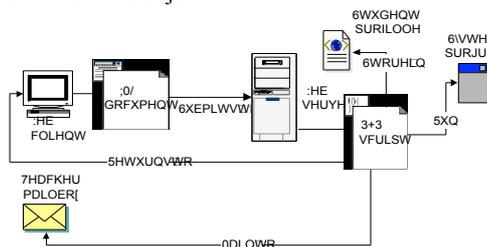


Figura 7. Esquema de RPEWA 2.

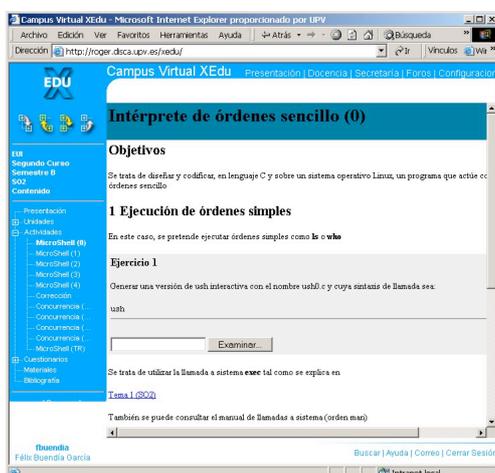


Figura 8. Ejemplo de presentación de la Unidad de Aprendizaje.

La Figura 8 representa una *Unidad de Aprendizaje* que es interpretada dentro del servidor Web mediante código PHP. Cada una de las unidades de aprendizaje lleva asociadas las siguientes acciones.

1. Presentación de una página HTML que contiene explicaciones teóricas procedentes de los elementos *paragraph* o formularios para la realización de actividades asociadas a los elementos *exercise*.
2. Actualización de los resultados obtenidos por el alumno en su perfil de estudiante. Dichos resultados pueden consistir en respuestas a cuestiones planteadas en la actividad o en la elaboración de programas como respuesta a los ejercicios propuestos. En este caso se proporciona la posibilidad de comprobar el funcionamiento de dichos programas, tal como se muestra en la Figura 9.
3. Notificación al profesor por correo electrónico de la realización de las actividades propuestas en la Unidad de Aprendizaje. De esta forma se pueden aplicar acciones de supervisión y evaluación tales como acceder a las respuestas sobre las actividades planteadas en la Unidad de Aprendizaje, así como obtener los archivos que representan los programas desarrollados por el alumno. Así, el profesor puede supervisar mediante una pantalla de la aplicación el trabajo del alumno.

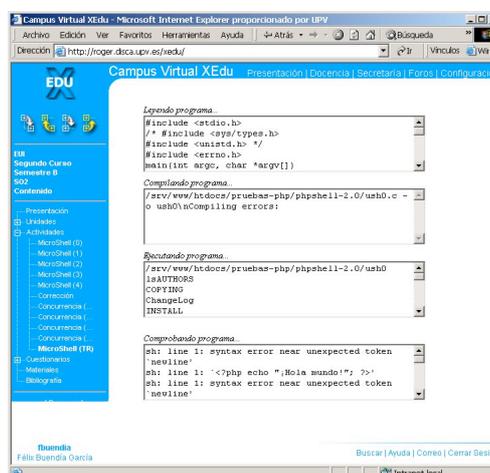


Figura 9. Ejemplo de ejecución de programa mediante RPEWA2.

4. Valoración sobre el uso docente de las herramientas

Las herramientas descritas previamente reflejan la contribución de tecnologías como simuladores y aplicaciones Web para facilitar la docencia en una disciplina fundamental en los *curricula* de Informática como son el estudio de los Sistemas Operativos.

La primera herramienta, consistente en un simulador de memoria virtual, se utilizó hace unos años en una asignatura de segundo curso. En dicha asignatura se impartían conceptos básicos de Sistemas Operativos tales como la gestión de procesos o de memoria, y el objetivo del simulador consistía en facilitar la comprensión de estos conceptos. También servía de apoyo para la realización de actividades de diseño consistentes en que el alumno elaborara un simulador propio aunque sin componentes gráficos. La experiencia de uso del simulador fue positiva y uno de los aspectos más valorados consistió en la posibilidad de visualizar el funcionamiento de algoritmos, tanto relacionados con la planificación de procesos como con la traducción de direcciones y asignación de espacio en memoria. No obstante, con la llegada de los nuevos planes de estudio en las titulaciones de informática de la Escuela de Informática se produjo una compresión de los contenidos básicos que pasaron de una duración

anual a una semestral. Ello significó una reducción del tiempo dedicado para el estudio de aspectos teóricos y asimismo de las actividades, como el uso del simulador, orientadas a facilitar su comprensión. Por tanto, dicha actividad quedó fuera de la planificación de prácticas y simplemente se dejó como trabajo opcional.

Como consecuencia de este cambio de orientación en los planes de estudio se dio cabida a una nueva asignatura de tipo semestral basada en el estudio de casos y de corte mucho más práctico. En este contexto se introdujeron actividades relacionadas con las herramientas que se describen en la sección 3. También se produjo una novedad en la forma de impartir algunas asignaturas y que consistía en la introducción de grupos denominados de “tele-enseñanza” (TEL) cuya finalidad era la de flexibilizar su docencia. Ambas cuestiones han contribuido a potenciar una serie de herramientas que aprovechan las posibilidades de la Web para acceder a las funciones y mecanismos de un sistema operativo.

Por ejemplo, en la asignatura de Sistemas Operativos se pasó de un grupo TEL con menos de cinco alumnos en el curso 2002-2003 a 29 alumnos en el curso 2003-2004 y 50 alumnos en el curso actual. Este aumento ha requerido el desarrollo de las herramientas utilizadas en dicho grupo, desde aplicaciones sencillas como RPEWA donde se produce una mera comprobación de programas C a través de una interfaz Web hasta la integración de dichas aplicaciones en un entorno de Campus Virtual como se manifiesta en la versión RPEWA2. El incremento del número de alumnos también ha exigido la incorporación de nuevas herramientas que faciliten la supervisión y control por parte del profesor.

El resultado de las experiencias llevadas a cabo durante el curso pasado junto con las ya iniciadas en el actual, ponen de manifiesto las siguientes cuestiones:

- Preparación: los alumnos no están acostumbrados a tomar iniciativas en el uso de herramientas que den soporte a la resolución de problemas que se plantean en las actividades prácticas. Habitualmente, ellos esperan un guión que les dirija la actividad a realizar.
- Conocimientos previos: se han detectado graves carencias en aquellos conocimientos (principalmente de programación) que son

requeridos para un aprovechamiento de las herramientas proporcionadas.

- Conexión con la teoría: en algunos casos se ha observado una dificultad en conectar el trabajo realizado a partir de las herramientas con los conceptos teóricos que subyacen a las actividades prácticas puestas en marcha.
- Evaluación: resulta compleja la interpretación de las actividades realizadas a partir del uso de las herramientas y su traducción a una nota que evalúe el trabajo del alumno. También se detectan casos de “plagiarismo” y resulta difícil reconocer el trabajo individual.

Todas estas cuestiones se han trasladado a un grupo de innovación docente que se ha creado recientemente en la Universidad Politécnica de Valencia (UPV) donde se debaten aspectos de aplicación de las nuevas tecnologías a la docencia. Dicho grupo de innovación se denomina NTA² [18] y está formado por personal docente de varios departamentos, interesados en las Tecnologías de la Información y la Comunicación (TIC) y sus aplicaciones docentes. Esta iniciativa forma parte de los llamados Proyectos de Adaptación al Espacio Europeo de Educación Superior (PAEEES) dentro de la UPV y está integrada por diversas acciones que van desde los sistemas de evaluación, la utilización de herramientas de campus virtual y trabajo colaborativo, el uso de laboratorios virtuales/remotos hasta el aprendizaje basado en proyectos. En este contexto se pretende analizar el uso de las herramientas descritas en el presente trabajo y evaluar sus futuras aplicaciones para fomentar el aprendizaje activo del alumno según las directrices del nuevo Espacio Europeo de Educación Superior.

5. Conclusiones

En este trabajo se han descrito dos ejemplos de herramientas que representan dos formas de enfocar las actividades prácticas en el ámbito docente de los sistemas operativos. También se ha indicado el contexto donde fueron utilizadas y las valoraciones de dicho uso, haciendo hincapié en la oportunidad que supone su incorporación en el ámbito del Espacio Europeo de Educación Superior.

Sin embargo, se han detectado las dificultades propias del desarrollo y mantenimiento de

herramientas que exigen una gran cantidad de recursos y dedicación. Ello exige que las instituciones académicas apuesten por este tipo de herramientas y contribuyan a su soporte y gestión. Por otro lado, la utilización de estas herramientas requiere una adaptación de las metodologías empleadas en la docencia de las materias relacionadas. En el contexto de los Sistemas Operativos, esto significa conceder un mayor protagonismo al trabajo individual de alumno y que el profesor dedique más esfuerzo a tutelar y supervisar las actividades realizadas con dichas herramientas. También son necesarios nuevos procedimientos de evaluación que tengan en cuenta la realización de estas actividades.

Referencias

- [1] A.B. Tucker et al., ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 1991, ACM Press; IEEE Comp. Soc. Press., 1991.
- [2] ACM/IEEE Task Force on the Year 2001 Model Curricula for Computing, Computing Curricula 2001(CC-2001), (Online: <http://www.computer.org/education/cc2001/>).
- [3] A. S. Tanenbaum and A. S. Woodhull, Operating Systems: Design and Implementation, 2ed., Prentice-Hall, 1997.
- [4] W A Christopher et a. (1993), The Nachos Instructional Operating System. Proceedings of the Winter 1993 Usenix Technical Conference, pp 481-489.
- [5] David Jones, Andrew Newman, A constructivist-based tools for operating systems education, Proceedings of EdMedia'2002, Denver, Colorado, Jun. 2002.
- [6] L.P. Maia, A.C. Pacheco, A Simulator Supporting Lectures on Operating Systems. 33rd ASEE/IEEE Frontiers In Education Conference 2003 Boulder, Colorado.
- [7] Jeff Magee and Jeff Kramer. Concurrency: State Models & Java Programs.
- [8] Llopis Mengual, J.Espinosa Rodilla. Simulador de gestión de memoria. Proyecto final de carrera. Universidad Politécnica de Valencia, 1999.
- [9] Innovations for Education in Information Technology through Multimedia and Communication Networks. Proyecto Socrates de Red Temática. (Online: <http://www.eui.upv.es/ineit-mucon/>).
- [10] HTML Shell (Online: <http://www.nmmm.nu/linux/sh/htsh.htm>).
- [11] M. Geisler. PHPShell (Online: <http://www.gimpster.com/wiki/PhpShell>).
- [12] Windows Script Host (WSH) (Online: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/wsconwshbasics.asp>).
- [13] J.M. Ruiz, J.F. Puchades. Programación de scripts Web. Proyecto final de carrera. Universidad Politécnica de Valencia, 2003.
- [14] F. Buendía, M Agustí, J.V Benlloch., E. Bisbal, M. Lluésma, XEDU, an Open Learning Management System Proposal. International Conference on Network Universities and E-learning 2003, Valencia (Spain).
- [15] F. Buendía, J.C. Burguillo, J.V Benlloch., D Rodríguez, J.M. Gómez, J.J. Vidal. Tools for Creation and Management of Didactic Resources in Electrical and Information Engineering. Computers and Education: Towards a Lifelong Learning Society, Kluwer Academic Publishers 2003.
- [16] Stüb,C., Kammerl, R., Freitag, B.A Teachware Management Framework for Multiple Teaching Strategies. In: Proceedings of ED-MEDIA 2000, World Conference on Educational Multimedia, Hypermedia & Telecommunications, Montreal, Quebec, 2000
- [17] F.Buendía, J.V Benlloch, J.M. Gomez, Development of Didactic Resources for Distance Learning based on Simulation, Computers and Education Towards an Interconnected Society, Ed. Kluwer Academic Publishers 2001, pp. 93-101.
- [18] F. Buendía et al. El Grupo de Innovación Docente NTA² “Nuevas Tecnologías para el Aprendizaje Activo”. Enviado al Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (Sintice 05).