UNIDAD 02: EVALUACIÓN DE PRODUCTO

PROPÓSITO DEL PROYECTO

Brindar al estudiante la oportunidad de interactuar con un sistema operativo real de tipo UNIX (XV6), diseñar y desarrollar extensiones simples a nivel de núcleo y espacio de usuario, y analizar el impacto de dichas extensiones sobre la gestión de memoria, planificación, sistema de archivos y seguridad básica, articulando los conceptos desarrollados en la Unidad II del curso (memoria virtual, planificación, E/S, archivos, sistemas distribuidos y seguridad).

RESULTADOS DE APRENDIZAJE

Al finalizar el proyecto, el estudiante será capaz de:

- Relacionar la teoría de paginación, memoria virtual y planificación con su implementación en un sistema operativo educativo.
- Implementar comandos y llamadas al sistema que accedan a información de memoria, procesos y archivos.
- Aplicar buenas prácticas de programación en C, control de versiones con Git y documentación técnica.

INTRODUCCIÓN

Los sistemas operativos son una de las piezas de software más complejas en computación ya que se encarga de multiplexar un limitado número de recursos computacionales entre diversos servicios, tareas y usuarios. Los sistemas operativos deben ser eficientes a la hora de compartir y asignar recursos, reduciendo los tiempos de espera experimentados por las aplicaciones y los usuarios, maximizando el uso de los recursos. Para alcanzar estos objetivos se deben diseñar mecanismos que aislen los procesos de modo que estos sientan que tienen acceso exclusivo de los recursos.

Para lograr esta separación adecuada entre las diferentes tareas o procesos que se ejecutan en un sistema de cómputo se han diseñado e implementado mecanismos que separan las operaciones ejecutadas por un proceso entre operaciones privilegiadas y no privilegiadas. Las operaciones privilegiadas son gestionadas y ejecutadas por el sistema operativo quien se debe encargar del acceso coordinado y eficiente de los recursos de hardware y de aquellas estructura de datos fundamentales que llevan un control y seguimiento de todas aquellas tareas y procesos que acceden a los recursos del sistema.

De otro lado, las operaciones no-privilegiadas tienen que ver con aquellas operaciones que no acceden ni a dispositivos de hardware ni a estructura de datos del sistema operativo pero que se orientan más al procesamiento de datos alfanuméricos y que son la razón de ser de los procesos de usuario, e.g. procesos que hacen búsquedas en bases de datos, secuenciamiento de genomas, análisis molecular, entre otros.

En este proyecto usted tendrá una primera aproximación a un sistema operativo real y para ello hará uso del sistema operativo XV6 el cual es un sistema operativo derivado de las primeras versiones del sistema operativo UNIX desarrollado por Dennis Ritchie, Brian Kernighan y Ken Thompson. Esta versión de UNIX se diferencia de la versión original fundamentalmente porque este sistema está orientado a arquitecturas de computadores x86 . Este sistema operativo es una versión muy limitada en cuanto a librerías y comandos disponibles en el sistema lo cual lo hace propicio para que el estudiante curioso pueda crear su propio sistema a partir de esta versión preliminar del sistema.

PREPARANDO EL AMBIENTE DE TRABAJO

El sistema operativo XV6 fue desarrollado en el MIT en el verano del 2006. El software se desarrolló con fines netamente académicos y provee el esqueleto básico de un sistema operativo funcional con la posibilidad de ser extendido. El software puede ser descargado a través de git y requiere de algunas herramientas de desarrollo adicionales que se describirán a continuación.

1. Requerimientos de software

Para llevar a cabo la descarga y compilación del software para su ambiente de trabajo se sugiere la ejecución de los siguientes comandos

```
sudo apt-get update
sudo apt-get install build-essential gdb git gcc-multilib qemu-system-x86
```

2. Descarga del sistema operativo
Una vez hayas descargado las herramientas de desarrollo podrás descargar el sistema operativo y posteriormente compilarlo.
El sistema operativo se puede descargar a través de la herramienta de versionamiento git. Ejecutando el siguiente comando tendrás una copia del sistema operativo en tu máquina.

```
git clone git://github.com/mit-pdos/xv6-public.git
cd xv6-public
```

3. Compilando el software

Una vez el software ha sido descargado, el proceso de compilación es bastante sencillo. En el directorio donde se clonó el software se debe ejecutar el siguiente comando:

```
make
```

Si todo sale bien, ud. debería ver algo como lo siguiente al final de la compilación.

```
dd if=/dev/zero of=xv6.img count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB) copied, 0.0191683 s, 267 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000903026 s, 567 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
351+1 records in
351+1 records out
180072 bytes (180 kB) copied, 0.000915411 s, 197 MB/s
```

En este momento ya se tiene una imagen del sistema operativo (xv6.img) disponible para ser usada. Para tener su sistema operativo en ejecución debe ejecutar, valga la redundancia, el siguiente comando:

```
make qemu
```

Evidencia: Captura de pantalla del sistema XV6 en ejecución (consola QEMU) con el prompt de usuario.

DESCRIPCIÓN DEL PROYECTO Y ENTREGABLES

El proyecto se desarrollará en grupos de dos (02) integrantes. Cada grupo extenderá XV6 mediante la implementación de, al menos, los siguientes componentes:

ENTREGABLE 1: Instrumentación de llamadas al sistema

- Modificar el código de XV6 para **registrar y mostrar por pantalla** las llamadas al sistema que se ejecutan.

- Para cada llamada al sistema, mostrar **el nombre de la syscall y los parámetros** usados en su invocación.
- Integrar esta funcionalidad de tal manera que pueda activarse mediante un comando o flag (para no saturar la consola permanentemente).

Archivos sugeridos a revisar: `syscall.c`, `sysproc.c`, `user.h`, `usys.S`.

Evidencia: Capturas de pantalla donde se observe la ejecución de comandos y el listado de llamadas al sistema asociadas.

ENTREGABLE 2: Comandos de usuario relacionados con la Unidad II

Implementar al menos dos (02) comandos nuevos o extendidos en XV6, orientados a los temas de Unidad II:

1. **Comando `uptime` extendido**
    - Mostrar el tiempo de ejecución del sistema (en ticks y/o en segundos).
    - Agregar información adicional sobre número de procesos activos o uso básico del CPU (por ejemplo, número de cambios de contexto).

2. **Comando de inspección de memoria o planificación** (por ejemplo `psmem` o `schedinfo`):
    - Mostrar información simple sobre:
        - Procesos y su estado (runnable, running, sleeping).
        - Contadores relacionados con cambios de contexto, colas del scheduler o uso de memoria por proceso (si es factible).

3. (Opcional pero recomendado) Comando relacionado con archivos, por ejemplo:
    - `lsx`: extensión de `ls` que muestre tamaño, número de enlaces u otra información de los inodos.
    - Un comando sencillo que verifique espacios libres/ocupados en el sistema de archivos.

Evidencia: Capturas de pantalla de la ejecución de los nuevos comandos, con casos de prueba representativos.

ENTREGABLE 3: Contador de invocaciones por llamada al sistema

- Diseñar una estructura de datos que mantenga, en el núcleo, el número de invocaciones que ha tenido cada llamada al sistema (p.ej., un arreglo indexado por el número de syscall).
- Implementar una nueva llamada al sistema que permita consultar esos contadores.
- Implementar un comando de usuario que:
    - Si recibe un identificador de syscall como parámetro, muestre únicamente el número de invocaciones de dicha llamada.
    - Si no recibe parámetros, muestre un cuadro resumen con todas las llamadas al sistema y su número de invocaciones.

Evidencia: Capturas de pantalla del comando mostrando el resumen global y la consulta individual.

ENTREGABLE 4: Informe técnico y repositorio Git

Cada grupo deberá presentar:

1. **Informe técnico en PDF** que incluya:
    - Portada (según formato de la Escuela).
    - Introducción y objetivos.
    - Descripción de las modificaciones realizadas (por sección: instrumentación, nuevos comandos, contador de syscalls).
    - Fragmentos relevantes de código comentado.
    - Resultados de pruebas (incluyendo capturas de pantalla ordenadas y referenciadas).
    - Conclusiones técnicas (relacionando los resultados con los contenidos de la Unidad II).

- Referencias bibliográficas (incluyendo el material de XV6 y la bibliografía básica del curso).
- Anexos (enlace al repositorio público)

2. **Repositorio público en GitHub**
   - Estructura clara del proyecto.
   - Historial de commits que refleje el trabajo colaborativo de los dos integrantes.
   - Archivo `README.md` indicando:
     - Integrantes.
     - Descripción breve del proyecto.
     - Instrucciones para compilar y ejecutar.

3. **Presentación oral (exposición de proyecto)**
   - Diapositivas (máx. 10) resumiendo: objetivo, arquitectura de cambios, demostración de comandos y conclusiones.
   - Demostración en vivo o video corto (3–5 min) mostrando el sistema en ejecución.

CONDICIONES DE TRABAJO Y ÉTICA ACADÉMICA

- El trabajo se realiza en grupos de dos estudiantes.
- Todo el código y la documentación deben ser originales del grupo; se permite el uso de ejemplos del libro de XV6 y del código base siempre que se citen adecuadamente.
- Cualquier coincidencia sustancial de código o informe entre grupos se considerará copia y se sancionará según el reglamento vigente de la Universidad.
- El código debe estar correctamente comentado y seguir una estructura clara para facilitar su lectura y mantenimiento.

REFERENCIAS

- "Xv6, a Simple Unix-like Teaching Operating System." 6.828 / Fall 2014. Accessed April 04, 2015. http://pdos.csail.mit.edu/6.828/2014/xv6.html.
- Junto al código fuente del sistema operativo,
  - book-rev8.pdf, describe algunos conceptos de los sistemas operativos vistos en clase pero profundiza en aquellos elementos que son importantes entender a la hora de implementar un sistema operativo del tipo UNIX.
  - xv6.pdf, este archivo contiene todo el código fuente del sistema operativo XV6 debidamente numerada en cada una de sus páginas. Es un documento complementario al pdf anterior.