EXOKERNEL

Semestre I-2014

Introducción - Exokernel

- ¿Qué define un SO?
  - Interfaz entre aplicaciones y recursos físicos.
- Ventajas de las interfaces.
  - Ya conocemos las ventajas.
    - ¿Cuáles son?

Introducción - Exokernel

- Desventajas de las interfaces.
  - Limitan el desempeño.
  - Limitan la libre implementación.
    - ¿De quién?
  - ¿A qué se deben estas desventajas?
    - Abstracción.
    - Portabilidad.
    - Características adicionales.

Introducción - Exokernel

- Filosofías tradicionales de SO.
  - Interfaces de acceso completas.
  - Manejo centralizado de recursos.
    - Manejo descentralizado.
  - Consideraciones
    - Conflictos → Aplicaciones.
    - Desempeño.
    - Flexibilidad.

Introducción - Exokernel

- ¿Cómo solventar dicho inconveniente?
  - Manejo distribuidos de los recursos por aplicaciones.
  - ¿Esto que ofrece?
    - Facilidad de expansión.
    - Facilidad en manejos específicos.
    - Posibilidad de remoción de ciertos componentes.

Introducción - Exokernel

- Enfoque tradicional
  - Ampliar el set de instrucciones del hardware.
  - Esconder información sobre los recursos vía abstracciones centralizadas.
  - Implementaciones particulares de la abstracción.
    - Procesos.
    - IPC.
    - Manejo de Interrupciones.
    - Sistema de Archivos.

Introducción - Exokernel

- Enfoque tradicional.
  - Abstracción centralizada.
    - "Máquina virtual" para aplicaciones.
      - ¿Qué brinda esta "máquina virtual"?
    - Implementaciones no removibles.
      - ¿Por quién?
      - Aplicaciones no confiables y/o no seguras.

Introducción - Exokernel

- Idea.
  - Es inaceptable el enfoque tradicional.
    - Negar a las aplicaciones las ventajas de dominarse en base a su definición de optimización.
    - Es restrictivo y poco flexible.
      - Definir nuevos recursos o abstracciones.

Introducción - Exokernel

- Evidencias.
  - No hablamos por hablar…
  - Apple et al.
    - Primitivas de propósito general para memoria virtual.
      - Almacenamiento persistente.
      - Recolector de basura.
      - Memoria compartida distribuida.

Introducción - Exokernel

- Evidencias.
  - Cao et al.
    - Manejo de caching de archivos en aplicaciones de alto nivel.
      - Reduce E/S alrededor de un 80%
  - Cheriton and Krueger.
    - Políticas especificas en memoria virtual.
      - Incremento en las prestaciones.

Introducción - Exokernel

- Evidencias.
  - Stonebraker.
    - Sistemas de archivos inapropiados.
      - Rendimiento de BD
  - Thekkath.
    - Retrasar el manejo de señales.
      - Reduce el costo de las excepciones en las aplicaciones.

Introducción - Exokernel

- Idea.
  - Plantear una arquitectura de un SO.
  - Abstracción del SO.
    - Implementada en niveles de aplicación.
    - Mediante software no confiable.

Introducción - Exokernel

- Idea.
  - Bajo el enfoque anterior.
  - Exokernel.
    - Multiplexación segura de recursos disponibles.
    - Protección.
    - Revocación.

Introducción - Exokernel

- Idea.
  - Usando el exokernel.
  - Aplicación.
    - Solicitar o tomar recursos.
    - Realizar manejo de eventos.
    - Participar en la revocación de recursos.

Introducción - Exokernel

- Idea.
  - Implementación de bajo nivel.
    - Implementación extremadamente eficiente.
  - Conjunto de Bibliotecas.
    - Bibliotecas del SO.
    - Trabajan sobre las interfaces del exokernel.
    - Implementan las abstracciones de bajo nivel.

Introducción - Exokernel

- SO con arquitectura exokernel
  - Aegis
  - ExOS

Introducción - Exokernel

- Prestaciones.
  - Mejor que cualquier kernel monolítico.
  - Reenvío de excepciones → 100 unidades.
    - Memoria Virtual en nivel de aplicación.
  - Manejo de IPC → 10 unidades.
    - Estructuras variadas.

Motivación - Exokernel

- El costo de abstracciones centralizadas.
  - Desempeño de las aplicaciones sufre.
    - ¿Por qué?
      - No existe una única manera de realizar abstracción de los recursos físicos.
      - No existe una única forma de implementar una abstracción centralizada de la mejor manera para todas las aplicaciones.

Motivación - Exokernel

- El costo de abstracciones centralizadas.
  - El SO esta forzado a elegir un soporte intermedio para las aplicaciones.
    - Por ejemplo:
      - Lecturas intensivas.
      - Escrituras intensivas.

End to End - Exokernel

- Abstracción centralizada → Demasiada generalidad.
- Intentar provee todas las características a las aplicaciones.
- ¿Consecuencias?

End to End - Exokernel

- **Lampson, Anderson, Massalin.**
  - Implementaciones de abstracciones centralizadas con un propósito general.
    - Forzar a las aplicaciones a sufrir sobrecarga.
- **Generalización.**
  - Importantes mejoras en el manejo del hardware a bajo nivel.
  - Aplicaciones de software más precisas y específicas.

End to End - Exokernel

- Las aplicaciones intentar conocer las operaciones del sistema y la interacción con el hardware.
  - ¿Con qué propósito?

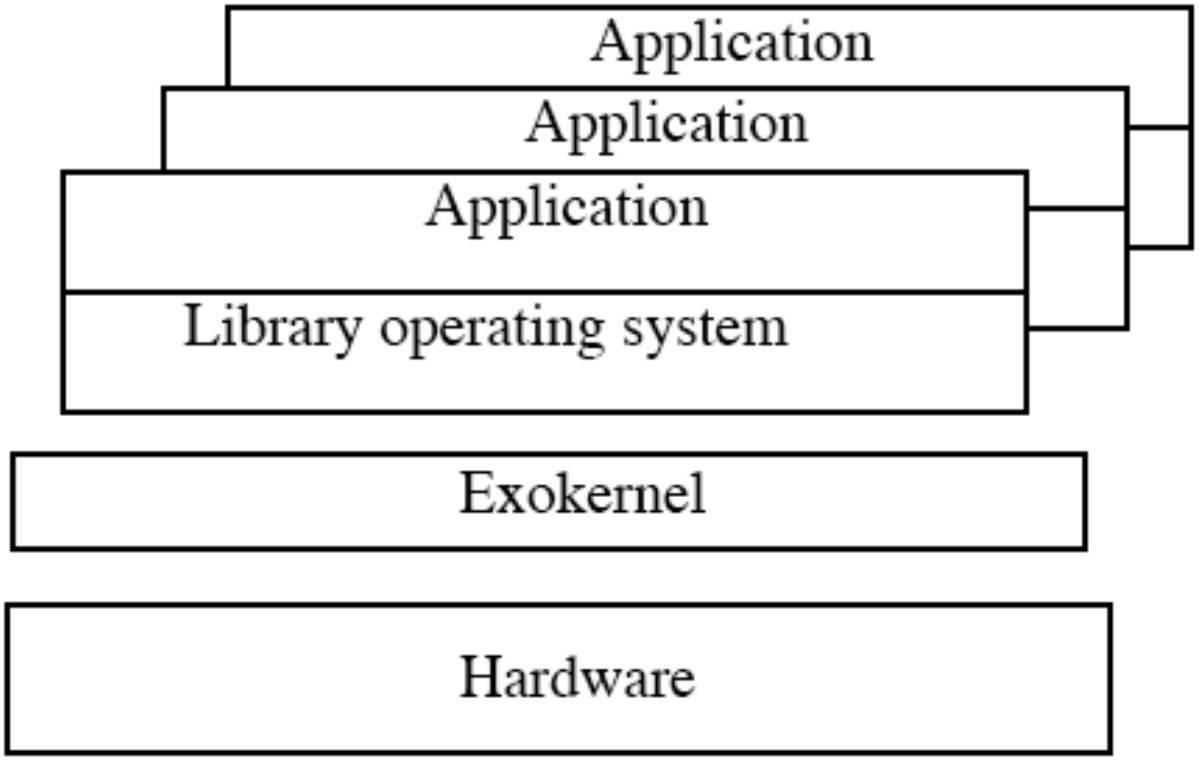
End to End - Exokernel

- Manejo de recursos de la forma apropiada.
- Toma de decisiones de acuerdo a la situación actual y no al caso común.
  - ¿Qué contradice lo anterior?

Arquitectura - Exokernel

- Pequeña capa de multiplexación de recursos.
- Bibliotecas.
  - “Sistemas Operativos” que implementan objetos y políticas.
- ¿Qué deberían preguntar?
  - Ideas.

Arquitectura - Exokernel



Bibliotecas - Exokernel

- Bibliotecas del Sistema Operativo
- Vienen a brindar
  - Flexibilidad
  - Eficiencia
- Implementaciones
  - Especializadas
  - Simples
  - Ejemplo

Bibliotecas - Exokernel

- Permite minimizar.
  - Cambios de modos o transiciones a kernel.
  - La mayoría del SO.
    - ¿Donde se ejecuta?
- No todo es bueno.
  - Problemas.
    - Portabilidad.
    - Complejidad.
  - Soluciones.
    - Ideas.

Bibliotecas - Exokernel

- ¿Alguien me obliga a utilizar las Bibliotecas?
- ¿Qué necesito para que esto funcione?
  - Soporte para Bibliotecas compartidas.
  - Soporte para enlace dinámico.

Bibliotecas - Exokernel

- ¿Cómo portar en un exokernel?
  - Emulando el código binario del SO y las aplicaciones.
  - Muevo las abstracciones necesarias al SO anfitrión.
  - Implemento de nuevo lo que deseo, tal vez recompilo y adapto.

Diseño - Exokernel

- Metas.
  - Aplicaciones extensibles.
  - Aplicaciones especializadas.
  - Reemplazo de las abstracciones centralizadas, con pie en el concepto de aplicaciones no confiables.
- Filosofía de diseño.
  - Control distribuido.

Diseño - Exokernel

- Responsabilidades.
  - Mapeo de recursos - usuarios (Propietarios).
  - Multiplexación segura de recursos.
    - Protección.
    - Puntos de servicio.
  - Revocación de acceso a los recursos.

Diseño - Exokernel

- No manejar los recursos más allá de ofrecer protección
  - Ejemplo:
    - Exportar instrucciones privilegiadas.
    - Exportar DMA.
    - Exportar los recursos de la máquina.
- Detalles de las interfaces.
  - Solicitar recursos.
  - Liberar recursos.
  - Usar recursos.

Diseño - Exokernel

- Principios guías.
  - Exponer el hardware.
    - Ubicación completa y granular.
  - Exponer nombres.
    - Espacios de nombres numerables.
  - Exponer eventos.
    - Revocación de recursos.

Diseño - Exokernel

- Multiplexación.
  - Dividir o No.
    - Ejemplo.
      - MIPS.
      - SPARC.
    - Costos.
- ¿Por qué exportar las instrucciones privilegiadas?
  - Abstracciones típicas del SO.
  - Encapsulamiento.
  - Verificación de recursos.

Diseño - Exokernel

- Nombramiento físico.
  - Manejo correcto y simple de los recursos.
  - Ejemplo:
    - # de páginas.

Diseño - Exokernel

- Multiplexación segura.
  - Tarea primordial.
  - Verificar privilegios.
    - Al usar el recurso.
    - Conocimiento limitado.
      - Bajo Nivel.
      - Alto Nivel → ¿Dónde se implementa?.
    - Unión segura.
      - Separar alto de bajo nivel.

Diseño - Exokernel

- Multiplexación Memoria Física.
  - Unión segura.
    - Página.
    - Propietario.
    - Capacidades.
- Multiplexación Frame Buffer.
  - Dificultad al conocer los dispositivos.
    - Ejemplos:
      - Disco.
      - Silicon Graphics.

Diseño - Exokernel

- Multiplexación de Red.
  - Múltiples protocolos.
    - Estudio del paquete.
    - No complicar el exokernel.

Diseño - Exokernel

- Revocación.
  - Recursos manejados a nivel de aplicación.
  - Reclamar recursos.
  - Invisible.
    - No involucrar a la aplicación.
    - Menor latencia.
    - Falta de control y escasez.
  - Visible.
    - Involucrar a la aplicación.
    - Ventajas.
    - Desventajas.

Diseño - Exokernel

- Protocolo de aborto.
  - ¿Qué pasa si la revocación falla?
  - Revocación en dos fases.
    - Ejemplo
  - ¿Sí falla también esto?
    - Opciones.
      - Mato todo.
      - Protocolo de aborto.
        - Recursos por la fuerza.
        - Vector de reposición.
        - Elección de recursos.

Exokernel Web Site

- http://pdos.csail.mit.edu/exo