

Introducción

Sistemas Operativos - ITIG

Álvaro Polo Valdenebro

apoloval@gsync.es

Febrero 2009



©2009 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike
disponible en <http://creativecommons.org/licenses/by-sa/2.1/es>



- 1 ¿Qué es un Sistema Operativo?
- 2 Un poco de historia
- 3 Estructura del Sistema Operativo

Contenidos

- 1 ¿Qué es un Sistema Operativo?
- 2 Un poco de historia
- 3 Estructura del Sistema Operativo

Un mundo sin sistemas operativos

- Imaginemos un mundo sin sistemas operativos
- Todo lo que tenemos es el hardware
- Pensad en cómo construir una aplicación

Un mundo sin sistemas operativos

- Imaginemos un mundo sin sistemas operativos
- Todo lo que tenemos es el hardware
- Pensad en cómo construir una aplicación
- ¿Cómo usamos los periféricos (disco, red, ...)?
- ¿Cómo gestionamos los recursos (CPU, memoria, disco, ...)?

Un mundo sin sistemas operativos

Ejemplo: lectura de disquetes

NEC PD765

- 16 comandos de control (leer, escribir, mover el cabezal, formatear pistas, inicializar, reiniciar, calibrar, ...)
- Lectura/escritura. 13 parámetros encapsulados en 9 bytes.
 - Dirección de bloque
 - Número de sectores por pista
 - Modo de registro
 - Tamaño de *gap* entre sectores
 - ...

Un mundo sin sistemas operativos

Ejemplo: lectura de disquetes

NEC PD765

- 16 comandos de control (leer, escribir, mover el cabezal, formatear pistas, inicializar, reiniciar, calibrar, ...)
- Lectura/escritura. 13 parámetros encapsulados en 9 bytes.
 - Dirección de bloque
 - Número de sectores por pista
 - Modo de registro
 - Tamaño de *gap* entre sectores
 - ...
- Eso no es todo
 - 23 códigos de estado y error en 7 bytes.
 - Encendido/apagado del servomotor
 - ...

Un mundo sin sistemas operativos

Ejemplo: lectura de disquetes (cont.)

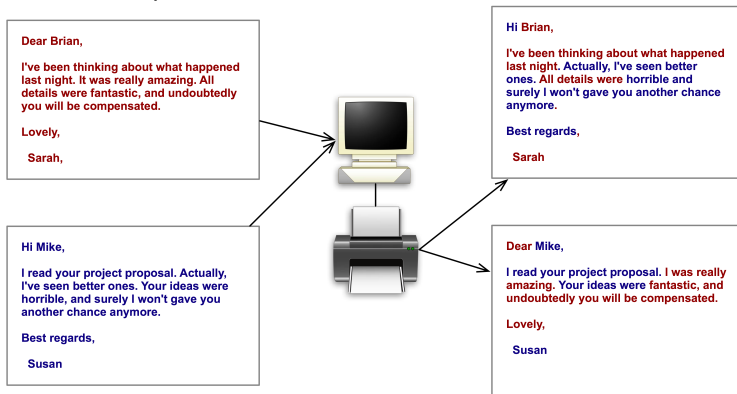
¿No es más fácil?

- Abrir fichero
- Leer/escribir fichero
- Cerrar fichero

Un mundo sin sistemas operativos

Ejemplo: Imprimir

Usar una impresora



Un mundo sin sistemas operativos

Al comienzo, el mundo era así

- Computación basada en *mainframes*
- Hardware muy **complicado** y muy **heterogéneo**
- Multitud de **recursos** que repartir

Un mundo sin sistemas operativos

Al comienzo, el mundo era así

- Computación basada en *mainframes*
- Hardware muy **complicado** y muy **heterogéneo**
- Multitud de **recursos** que repartir

En aquel contexto

- Queremos **abstraernos** de la complejidad y heterogeneidad del hardware
- Queremos **abstraernos** de la gestión de recursos

Un mundo sin sistemas operativos

Al comienzo, el mundo era así

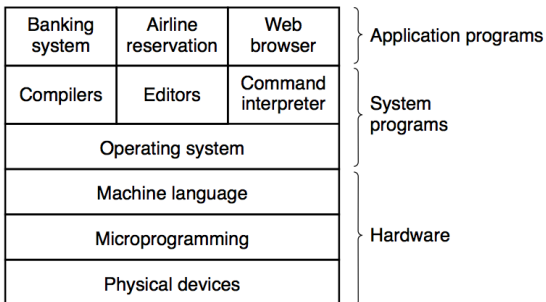
- Computación basada en *mainframes*
- Hardware muy **complicado** y muy **heterogéneo**
- Multitud de **recursos** que repartir

En aquel contexto

- Queremos **abstraernos** de la complejidad y heterogeneidad del hardware
- Queremos **abstraernos** de la gestión de recursos
- Queremos **software de sistema**

Software de Sistema

Creamos una capa sobre el hardware que nos **abstraiga**



Software de Sistema

Ya no tenemos hardware, tenemos una **máquina virtual**

- Ya no hay disco, hay **ficheros**
- Ya no hay red, hay **sockets**
- Ya no hay CPU, hay **procesos**
- Ya no hay memoria, hay **memoria virtual** y **heap**

En resumen...

Creamos sistemas operativos para

- **Abstraernos** de la complejidad del hardware
- **Abstraernos** de la heterogeneidad del hardware
- **Abstraernos** de la complejidad de la gestión de recursos

Contenidos

- 1 ¿Qué es un Sistema Operativo?
- 2 Un poco de historia
- 3 Estructura del Sistema Operativo

Primera generación de computadoras

El comienzo, 1945-55: calculadoras de sala

- Tubos de vacío, interruptores, "lucecitas"...
- Menos potentes que una calculadora de bolsillo
- Un mismo grupo diseña, construye, programa, opera y mantiene "su" máquina.
- Programación en lenguaje máquina cableando los programas. Aparecen las tarjetas perforadas.
- Aplicaciones: cálculo numérico sencillo, tablas de senos y cosenos...

Primera generación de computadoras

El comienzo, 1945-55: calculadoras de sala

- Tubos de vacío, interruptores, "lucecitas"...
- Menos potentes que una calculadora de bolsillo
- Un mismo grupo diseña, construye, programa, opera y mantiene "su" máquina.
- Programación en lenguaje máquina cableando los programas. Aparecen las tarjetas perforadas.
- Aplicaciones: cálculo numérico sencillo, tablas de senos y cosenos...
- Sin sistema operativo
- Sistemas monousuario/monoproceso
- Desperdicio de recursos

Segunda generación de computadoras

Transistores y procesamiento por lotes, 1955-65:

- Aparece el transistor a mediados de los años 50.
- Comienzan a comercializarse
- Los roles (diseñador, operador, programador) se especializan
- Procesamiento por lotes
 - Las tarjetas perforadas se vuelven en cintas de entrada (lotes)
 - Las cintas de entrada se leen y se ejecutan y el resultado se escribe en cintas de salida
 - El resultado se lee de la cinta de salida y se imprime

Segunda generación de computadoras

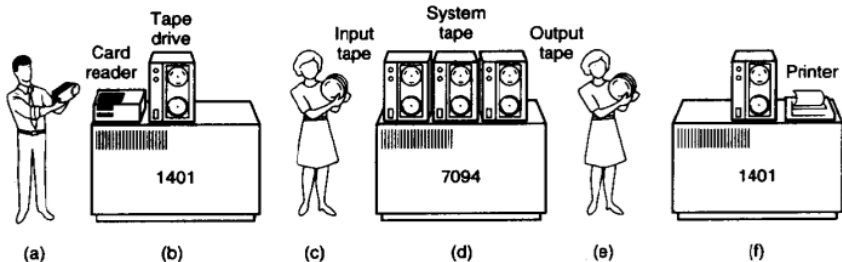
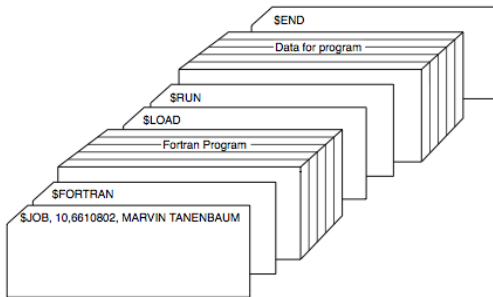


Figure 1-2. An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

Segunda generación de computadoras

- Tarjetas de control (\$JOB, \$FORTRAN \$RUN, ...)
- Un monitor dirige el proceso por lotes
- Reutilización de librerías (bloques de tarjetas)



Segunda generación de computadoras

- Formas primitivas de **sistemas operativos**
- Líneas de máquinas incompatibles entre sí
- Problemas de escalabilidad
- Desperdicio de recursos

Tercera generación de computadoras

Multiprogramación, 1965-80

- Circuitos integrados
- Familias de máquinas (IBM System/360)
- Compatibilidad binaria
- Sistemas de propósito general
- Con sistema operativo (OS/360)

Tercera generación de computadoras

Problemas de la segunda generación

- Las aplicaciones comerciales desperdician CPU (80-90 % E/S)

Multiprogramación

- Varios trabajos en memoria a la vez (particiones de memoria)
- Hardware de protección del entorno del proceso
- Planificador: decide qué trabajo ejecutar

Spooling (*Simultaneous Peripheral Operation On-Line*)

- Los trabajos se leen de las tarjetas y se guardan en disco
- Cuando una partición queda libre, se escoge un nuevo trabajo de disco

Tercera generación de computadoras

Usuarios descontentos.

- La CPU se comparte, pero las tareas tardan demasiado en completarse.
- Si una tarea falla, hay que esperar a que acabe el resto y volver a empezar

Timesharing, o tiempo compartido

- Los trabajos se asocian a un terminal
- El usuario típico demanda CPU, pero “en picos”.

Tercera generación de computadoras

Aparecen sistemas de tiempo compartido

- CTTS - *Compatible Time-Sharing System*
 - Problema: hardware de protección de memoria
- MULTICS - *MULTiplexed Information and Computing Service*
 - Ideado como "servicio" al estilo de la red eléctrica
- UNIX
 - Adaptación de MULTICS para sistemas DEC PDP-7 (minicomputadoras)
 - Múltiples implementaciones incompatibles
 - Surge el estándar POSIX

Cuarta generación de computadoras

Ordenadores personales (PCs), 1980 - actualidad

- LSI - *Large Scale Integration*
- Nace el ordenador personal => monousuario
- MS-DOS - *Microsoft Disk Operating System*
- MS-Windows
- UNIX evoluciona, especialmente en *workstations*
 - BSD - *Berkeley Software Distribution*
 - Linux

El presente - ¿quinta generación?

The network is the computer, 1990 - hoy

- Los ordenadores están interconectados - Internet
- Los recursos se comparten (impresoras, ficheros, CPU, ...)
- Middleware
- Arquitecturas Orientadas a Servicios (SOA)
- Cloud computing
- Plan9, Inferno

Contenidos

- 1 ¿Qué es un Sistema Operativo?
- 2 Un poco de historia
- 3 Estructura del Sistema Operativo**

El núcleo del sistema operativo

Núcleo del sistema operativo o **kernel**

- Ofrece la abstracción básica del hardware a través de **llamadas al sistema**
- Se ejecuta en **modo supervisor**

Modos de operación de la CPU

- **modo supervisor** o kernel. Control total sobre el hardware
- **modo usuario**. Acciones limitadas (E/S, MMU, ...)
- Pasamos de modo supervisor a usuario cambiando un *flag* del procesador
- Pasamos de modo usuario a supervisor con **interrupciones**
 - **software** ==> llamada al sistema
 - **hardware** ==> el núcleo recupera la CPU

Tipos de núcleo

- Núcleos **monolíticos**
 - Todos los servicios (ficheros, red, ...) se ofrecen desde el núcleo
 - Ventaja: es sencillo de implementar (drivers en modo kernel)
 - Desventaja: si algo falla, la máquina se viene abajo
 - Ejemplo: Linux, Windows, ...
- **Microkernels**
 - El núcleo solo ofrece servicios básicos (procesos, IPC, memoria)
 - El resto, se ejecuta en espacio de usuario (servidores)
 - Ventaja: más estable
 - Desventaja: bajo rendimiento, más complejo de implementar
 - Ejemplo: Mach kernel
- Núcleos **híbridos**
 - Algunos servicios esenciales (red, ficheros) se implementan en el núcleo
 - El resto, como servidores
 - Ejemplo: XNU (NEXTSTEP, Mac OSX)