Introdução aos Sistemas Operativos

01/03/11

Objectivos

- Reconhecer um Sistema Operativo (SO).
- Identificar os diferentes tipos de SOs.
- Saber distinguir os componentes base de um SO.
- Reconhecer a importância do kernel num SO.

O que é um Sistema Operativo?

- Conjunto de programas, código e dados que serve de intermediário entre o utilizador e o hardware.
- Esconde do utilizador os pormenores do hardware.
- Controla a execução de aplicações.
- Efectua a gestão de memória, periférios, rede e restantes recursos.
- Sistemas da família Windows ™, GNU Linux e OS X para computadores pessoais, Android ™ e iOS ™ para dispositivos móveis.

Onde se encontra?

Necessidade de um SO!

- Facilitar a utilização de um computador (como seria dirigir manualmente os milhares de circuitos electrónicos?).
- Rentabilizar os recursos de hardware.
- Ponto central de comunicação, coerente e constante.
- Fiabilidade e segurança de dados garantindo que o utilizador apenas faz o que é permitido (a segurança/fiabilidade é um processo complexo).

O que nos oferece

- Facilidade no desenvolvimento de aplicações novas.
- Execução das aplicações (jogos, produtividade, programação, etc.).
- Acesso aos dispositivos de E/S (discos, placas de rede, teclados, ratos, etc.).
- Acesso facilitado e controlado aos ficheiros de informação.
- Limitações de acesso ao próprio sistema (protecção dos recursos através de login).

O que nos oferece

- Tratamento de erros (numa primeira fase impedimento de erros) e recuperação em situações fatais:
  - Erros internos e externos de harware
  - Erros de memória
  - Falhas em dispositivos de E/S
  - Erros de software
- Contabilização de recursos através de estatísticas.
- Muito mais!

Exemplo: Acesso ao site www.silora.com

1. Utilizador abre o browser (rato ou comando).
2. O SO verifica que recursos são necessários para a execução.
3. Se existirem recursos suficientes os mesmos são alocados ao browser.
4. Se não existirem recursos suficientes o SO tenta obter recursos removendo-os de outras aplicações não usadas (paging, swap, etc).
5. O browser tenta aceder ao endereço através dos vários protocolos necessários, pelo caminho o SO recebe as interrupções referentes aos acessos à rede e executa as tarefas associadas.
6. Durante o processo outras aplicações entram em jogo, e o SO faz a gestão das mesmas, dos recursos e especialmente da utilização de CPU.
7. O browser mostra o resultado ao utilizador.

Sistemas Operativos Iniciais

- Máquinas que ocupavam salas.
- Necessitavam de operadores especializados, responsáveis também pela sua programação.
- A execução de um programa exigia uma preparação complexa e morosa.
- A execução era acompanhada através de paineis luminosos.
- O resultado tinha de ser guardado em fitas ou impresso.

Sistemas Operativos batch – anos 50

- Processamento de programas em lotes.
- Melhor rentabilidade através da diminuição do tempo de carregamento e controlo.
- Já utilizavam processador.
- Tinham acesso a operações de E/S.

Sistemas com monitor residente

- Faziam uso de um programa que estava encarregado de:
  - Controlar a execução de um programa.
  - Carregar módulos quando necessário.
  - Passar os recursos para o próximo programa que deles necessitava.
- Cada programa tinha de incluir instruções que indicavam as acções que o monitor deveria executar.

Sistemas com operações E/S

- As operações de E/S são sempre demoradas (comparando com o tempo que demora um ciclo de CPU).
- Se forem realizadas em série, uma após a outra, a execução dos programas provoca um desaproveitamento do CPU bastante elevado.
- Para resolver este problema podemos usar:
  - Spooling (Simultaneous Peripheral Operation On-line), ex: impressora.
  - Processamento de E/S off-line.

Sistemas com processamento off-line de E/S

- Os programas são lidos por um dispositivo próprio, que os guarda em fita magnética.
- O CPU obtém os dados a partir da fita magnética.
- A velocidade ganha aumenta a performance de execução.

Spooling

- SPOOL - Simultaneous Peripheral Operation On-Line.
- Substituição das fitas magnéticas por discos magnéticos, passando-se de acesso sequencial para acesso aleatório.
- O disco torna-se um buffer que armazena os dados de E/S.
- Exemplo mais conhecido: spooler da impressora.

Batch Multi-tarefa

- Vários programas em disco são carregados para memória
- O SO alterna entre os programas definindo o tempo de execução de cada um.
- Para decidir o SO tem em consideração os recursos disponíveis.
- Vários programas concorrem para utilizar o CPU.
- Só se torna eficaz se os programas residirem em memória.
- O uso de memória traz consigo a desvantagem do swapping.
- Nem todos os programas podiam ter swapping aplicado.

Sistemas multi-tarefa

- Evolução lógica dos sistemas de batch.
- Os processos alternam frequentemente.
- Existe comunicação com o utilizador, o sistema torna-se verdadeiramente interactivo.
- A interactividade com o utilizador traz:
  - Sistemas de ficheiros virtuais.
  - Necessidade de processamento de texto.
  - Linguagens de interacção com o sistema.

Requisitos multi-tarefa

- A nível de hardware:
  - Interrupções E/S e DMA.
  - Executar instruções quando os dispositivos de E/S estiverem ocupados.
  - Gestão de memória dado que as instruções têm de residir em memória.
  - Protecção da memória
- A nível de software:
  - Escalonamento, qual dos programas deve executar.
  - Gestão de recursos e limitação de acesso, dois programas não devem aceder ao mesmo recurso.

Sistemas de time-sharing

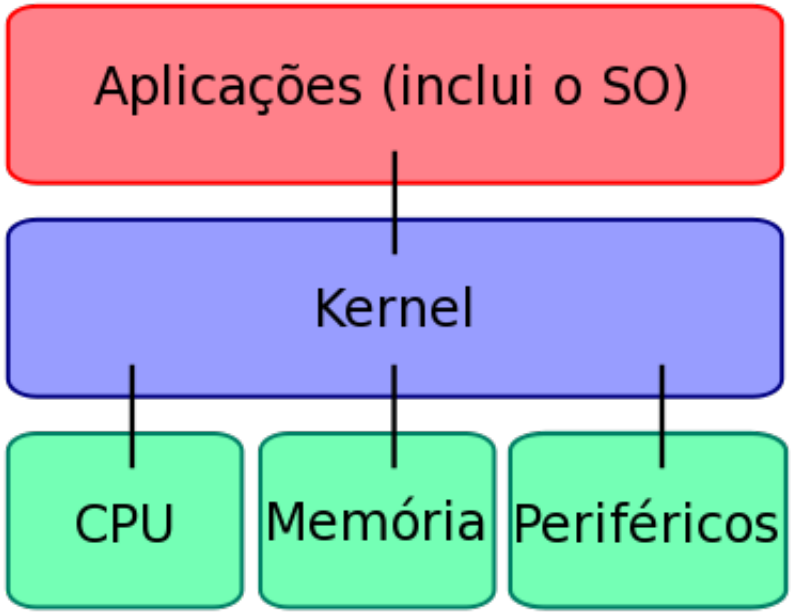
- Sistemas multi-tarefa para suportar várias tarefas.
- Diferentes de sistemas batch que não suportam interactividade.
- Tempo de CPU partilhado.
- Vários utilizadores, várias aplicações, pedidos que nunca terminam
- Chegámos à actualidade.

As possibilidades são imensas

- Sistemas em tempo real (NASA).
- Servidores com sistemas próprios.
- Dispositivos móveis.
- Frigoríficos, torradeiras, etc.

kernel

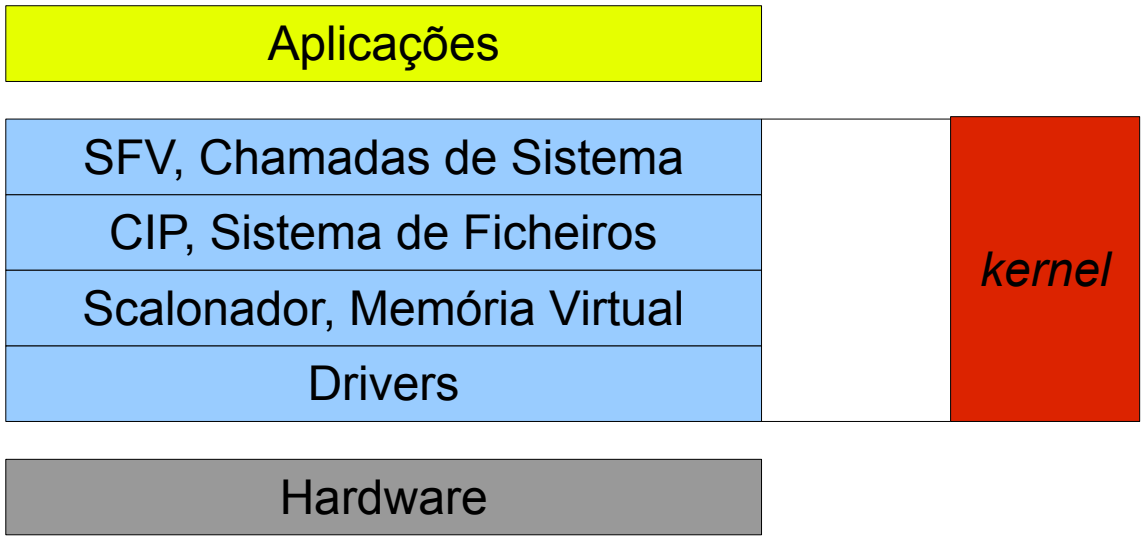
- Coração do SO.
- Faz a ponte entre o hardware e o processamento de dados
- Conjunto de funcionalidades mínimas, necessárias controlar, para que uma aplicação funcione correctamente.



kernel

- Consequência do desenvolvimento em camadas.
- Complexo de desenvolver
- Primeira abstracção do hardware
- Gestão dos hardware mais importante (CPU, E/S)
- Implementações variadas.

Kernel Monolítico



Resumindo

- O sistema operativo é necessário para que seja possível a interacção com o computador
- É responsável pelo controlo de todos os recursos.
- Sofreram uma evolução desde simples cartões perforados até sistemas complexos que executam em telemóveis
- O seu componentem mais importante é o kernel.
- Os sistemas que iremos usar possuem um kernel monolítico.

Conclusão

- Sem um sistema operativo estaríamos condenados a utilizar o computador através de botões e paineis luminosos.

Créditos

- Imagens obtidas na Wikipedia, licenciadas sob CC, ou criadas pelo formador.
- Apontamentos da cadeira de Sistemas Operativos, ESTG IPL.
- Formador Sérgio Lopes.