

INTRODUÇÃO

- O que é um Sistema Operativo (SO) ?
- Objectivos de um SO.
- Alguns conceitos básicos sobre SO's.
- Evolução dos SO's. Tipos de SO's.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

O que é um Sistema Operativo ?

Um programa grande e complexo (/ conjunto de programas) que controla a execução dos programas do utilizador e actua como intermediário entre o utilizador de um computador e o *hardware*.

Objectivos principais de um SO:

- ♦ fornecer uma gestão eficiente e segura dos recursos computacionais (gestão + controlo)
- ♦ fornecer ao utilizador uma máquina virtual mais fácil de programar do que o *hardware* subjacente (conveniência + eficiência)

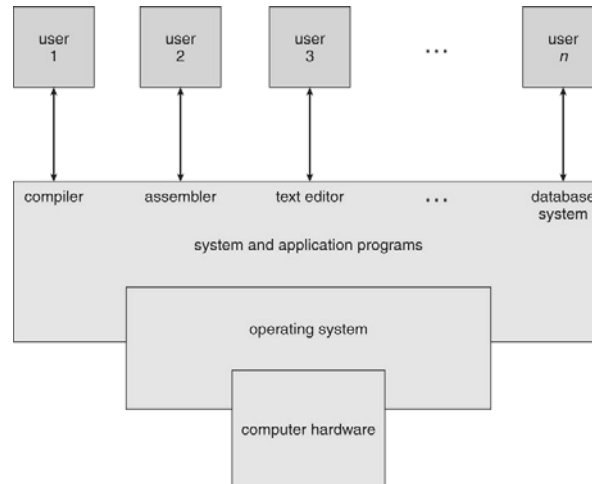
O que seria dos programadores sem um sistema operativo ?



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Onde "encaixa" o SO ?



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Alguns conceitos sobre SO's

Núcleo (*Kernel*)

- ♦ A parte principal do SO. Contém código para os serviços fundamentais. Está sempre em memória principal.

Device Drivers

- ♦ Código que fornece uma interface simples e consistente com os dispositivos de I/O
- ♦ Podem fazer parte do *kernel* ou não.

Programa

- ♦ Um ficheiro do disco contendo código numa linguagem de alto nível ou código-máquina (programa executável).

Processo

- ♦ Um programa em execução.
- ♦ A colecção de estruturas de dados e recursos do SO detidos por um programa enquanto está a ser executado.



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Ficheiro

- ◆ Colecção de informação relacionada entre si.
- ◆ Unidade lógica de armazenamento.
- ◆ O SO mapeia os ficheiros em dispositivos físicos onde a informação é gravada de forma permanente (memória secundária).
- ◆ Para muitos utilizadores, o sistema de ficheiros é o aspecto mais visível de um SO.

Chamadas ao sistema

- ◆ Os programas do utilizador comunicam com o SO e pedem-lhe serviços fazendo chamadas ao sistema.
- ◆ A cada chamada corresponde uma rotina da biblioteca de sistema.
- ◆ Esta rotina coloca os parâmetros da chamada ao sistema em locais especificados (ex.: registos do processador) e executa uma instrução de trap para passar o controlo ao sistema operativo.

Shell

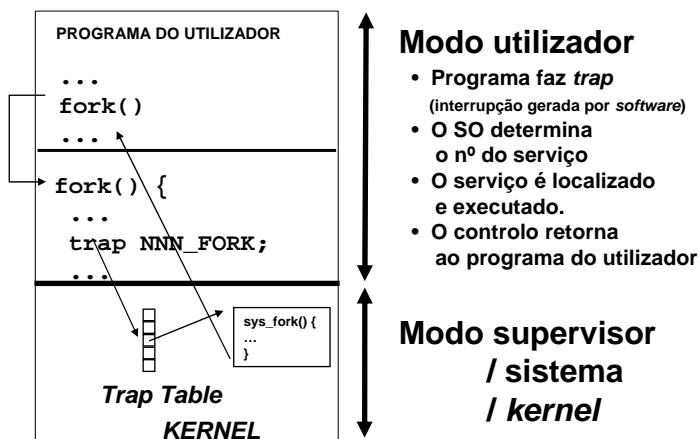
- ◆ Interpretador de comandos dados ao sistema operativo.
- ◆ Frequentemente, não faz parte do sistema operativo.



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

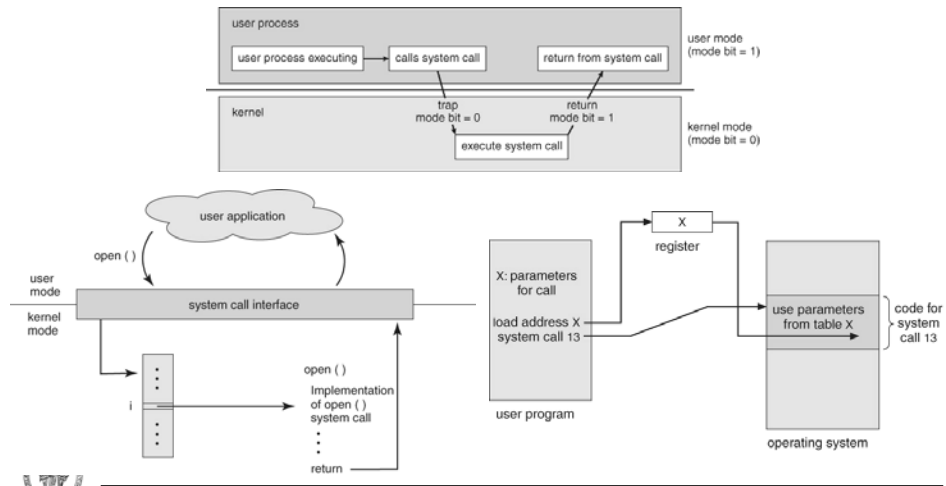
Chamadas ao sistema

FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Chamadas ao sistema



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Evolução dos sistemas operativos

- Processamento em série
- Processamento em lote (*batch*)
- Multiprogramação
- Tempo partilhado (*time-sharing*)
- Multiprocessamento
- Sistemas distribuídos
- Sistemas de tempo-real



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Evolução ...

- Processamento em série
 - Processamento em lote (*batch*)
- ... melhoramentos sucessivos
- dispositivos de *I/O*
 - implementação dos SOs
-
- evolução do *SW* ↔ evolução do *HW*



FEUP

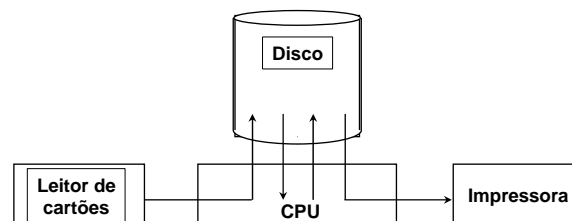
MIEIC
Faculdade de Engenharia da Universidade do Porto

Melhoramentos:

- ♦ Sobreposição das operações de entrada e de saída (*I/O*)
(coincide c/ a introdução de canais de *DMA*, controladores de periféricos, ...)

Spooling (*S**imultaneous* *P**eripheral* *O**perations* *O**n-L**i**n**e*)

- ♦ Forma de *buffering*: usar discos p/guardar temporariamente as *I/O*s.
- ♦ Permite sobrepor a fase de cálculo de um processo c/ a fase de *I/O* de outro.

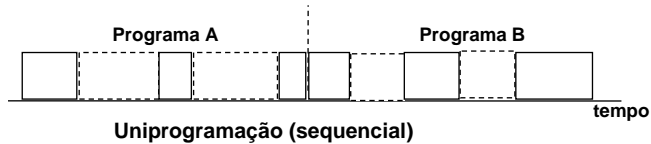


FEUP

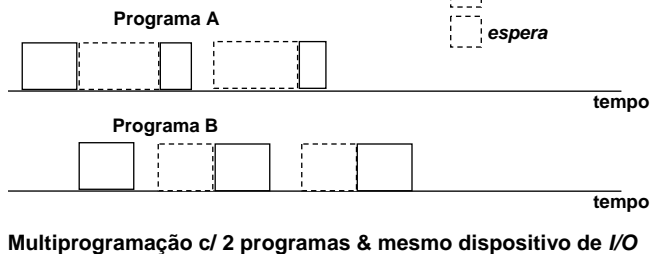
MIEIC
Faculdade de Engenharia da Universidade do Porto

Multiprogramação

Várias tarefas são mantidas em memória simultaneamente, e a *CPU* é partilhada entre elas



Quando o programa actual fica à espera que uma operação de *I/O* (*p/mesmo dispositivo*) se complete, o processador pode executar outro programa



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Multiprogramação

Multiprogramação: execução interlaçada de processos.

Algumas características do SO necessárias para multiprogramação

- ♦ Escalonamento da *CPU*
o sistema deve escolher entre os vários processos prontos a executar, aquele que vai ser executado
- ♦ Gestão de memória
alocar a memória aos diferentes processos
- ♦ Gestão de *I/O*
controlar o acesso aos dispositivos de *I/O*
- ♦ Protecção
não deve haver possibilidade de os processos se afectarem mutuamente em todos os níveis: escalonamento de *CPU*, memória acessível, *I/O*, ...



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Sistemas de tempo partilhado (*time-sharing*) - Computação interactiva

- ♦ Vários utilizadores simultâneos, cada um com a impressão de que tem o computador só para si.
- ♦ A *CPU* é partilhada entre diversas tarefas que são mantidas em memória e em disco (a *CPU* só é alocada a uma tarefa se ela estiver em memória).
- ♦ A comutação entre tarefas ocorre com uma elevada frequência.
- ♦ É possível a comunicação *on-line* entre o utilizador e o sistema;
- ♦ Deve existir um sistema de ficheiros *on-line* para que os utilizadores possam aceder aos programas e aos dados.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Outras evoluções

- Computadores pessoais
- Sistemas embebidos
- Sistemas de tempo-real
- Redes de computadores
- ...

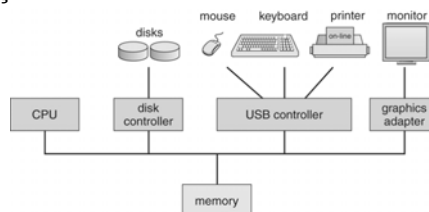


FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

HARDWARE DE UM SISTEMA DE COMPUTAÇÃO

- ♦ Revisão de conceitos básicos sobre *hardware*
 - » Processador (*recordar conceitos de Arq.Comp.*)
 - » Interrupções
 - » Processamento de E/S
 - » Memória
- ♦ Protecção do *hardware*



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Interrupções

- Uma **interrupção** é um mecanismo que permite que o processamento normal de um processador seja interrompido.
- As interrupções são usadas para aumentar a eficiência, especialmente quando se usam componentes que operam a velocidades diferentes.
- Permitem que enquanto decorre uma operação de E/S de um processo o processador continue a executar outros processos
- → base da multiprogramação
- Alternativa ao uso de interrupções: *polling*
 - ♦ 😞 introduz busy-waiting



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Acesso Directo à Memória (DMA)

Necessário um controlador de DMA ligado ao barramento do sistema.

Quando é necessário fazer E/S

- ♦ o processador informa o controlador do dispositivo de E/S do que pretende fazer e onde está ou vai ficar a informação a transferir
- ♦ o processador continua a executar outras instruções
- ♦ o dispositivo de E/S transfere a informação directamente de/para a memória
- ♦ quando o DMA termina é gerada uma interrupção

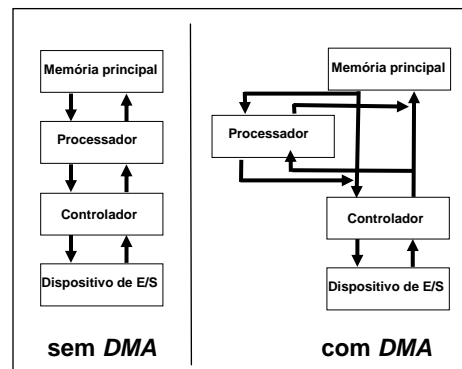
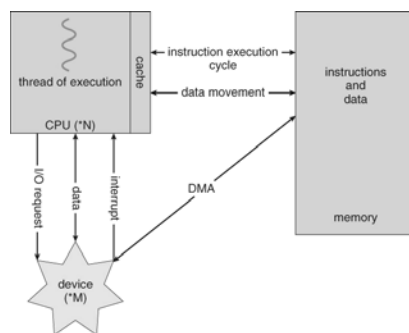


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Entrada / saída



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Protecção do *hardware*

Aumentar a utilização do sistema

- ⇒ partilha do sistema
- ⇒ vários programas a executar em simultâneo
- ⇒ protecção

O SO deve impedir que
um programa incorrecto ou "mal intencionado"
impeça os outros programas de executar.

Alguns erros de programação são detectados pelo *hardware*.

Normalmente estes erros são tratados pelo SO.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Protecção do *hardware* (cont.)

Protecção

- ♦ duplo modo de operação
 - » modo utilizador
 - » modo supervisor / sistema / monitor / privilegiado (instruções privilegiadas)
- ♦ protecção de E/S's
 - » os utilizadores não conseguem fazer E/S directamente, só através do SO
- ♦ protecção da memória
 - » protecção da área de mem. do SO e dos utilizadores feita por registos especiais
- ♦ protecção do processador
 - » temporizador impede que uma aplicação tome conta do processador indefinidamente



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Requisitos de *hardware* para multiprogramação

Um SO com multiprogramação necessita de suporte de *hardware*:

- ◆ temporizador
- ◆ *hardware* de DMA
- ◆ mecanismo de interrupções com prioridades
- ◆ duplo modo de operação do processador
- ◆ mecanismo de protecção da memória
- ◆ mecanismo de atribuição dinâmica de endereços
- ◆ ...



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

ESTRUTURA DO SISTEMA OPERATIVO

- Componentes do sistema operativo
- Tipos de estrutura
- Estrutura de sistema operativos concretos



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Componentes do S.O.

Pontos de vista de um sistema operativo:

- ♦ serviços que fornece
- ♦ interface que disponibiliza p/ utilizadores e programadores
- ♦ seus componentes e interligações

Componentes do sistema operativo:

- ♦ Gestão de processos
- ♦ Gestão da memória principal
- ♦ Gestão da memória secundária
- ♦ Gestão de ficheiros
- ♦ Gestão de entradas/saídas
- ♦ Gestão de rede
- ♦ Sistema de protecção/segurança

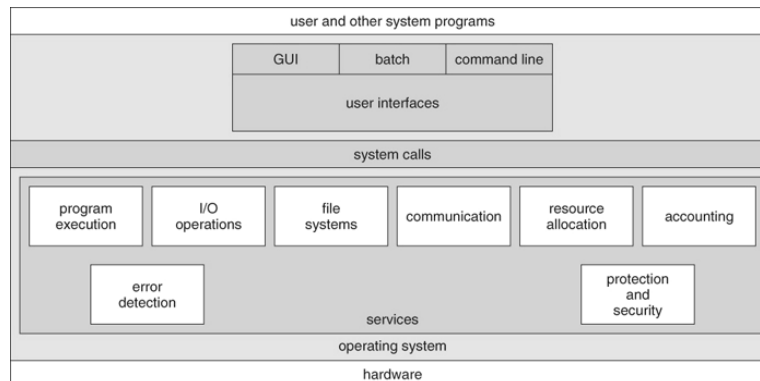


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Serviços fornecidos por um SO



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Tipos de estrutura de um S.O.

Estrutura:

- ◆ monolítica
- ◆ em camadas
- ◆ *microkernel*



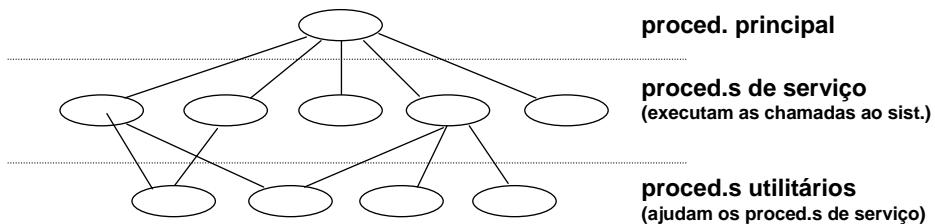
FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Estrutura monolítica

Primeiros S.O.'s.

- Não há estruturação ...
 - ◆ o S.O. é escrito como um conjunto de procedimentos cada um dos quais pode chamar qualquer outro.
- ... ou há uma pequena estruturação (ex: MS-DOS):



Dificuldades da estrutura monolítica:

- ◆ difícil de compreender
- ◆ difícil de modificar
- ◆ pouco fiável (um erro "em qualquer lado" pode provocar um *crash*)



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Estrutura em camadas

- O S.O. é dividido num certo número de camadas (níveis) cada qual construída por cima da anterior.
 - ♦ camada de mais alto nível - interface com o utilizador
 - ♦ camada 0 – *hardware*
- Sistema operativo modular
 - ♦ Para cada camada especificar a funcionalidade e as características.
 - ♦ É possível alterar a estrutura interna de cada camada desde que a interface com as outras camadas se mantenha inalterada.
 - ♦ Cada camada só usa funções e serviços das camadas inferiores.
 - ♦ Uma camada não necessita de "saber" como as operações da camada inferior são implementadas, mas apenas o que elas fazem.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Dificuldades da estruturação em camadas:

- ♦ Definição adequada das camadas
 - » porque cada camada só deveria poder usar as funções do nível inferior, mas ...
 - » ... ex: o sistema de gestão de ficheiros deveria ser um processo numa camada superior à de gestão de memória virtual; por sua vez, esta deverá poder usar ficheiros (!)
- ♦ Tende a ser menos eficiente do que outros tipos
 - » ex: para um programa do utilizador executar uma operação de I/O
 - executa uma chamada ao sistema
 - que faz um *trap* à camada de I/O
 - que chama a camada de ...
 - ... até chegar ao *hardware*

Os sistemas são frequentemente modelados como estruturas em camadas mas nem sempre são construídos dessa forma.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Estrutura baseada em *microkernel*

- Tendência nos S.O.'s modernos:

- ♦ Deslocar código para as camadas superiores deixando um *kernel* mínimo.
- ♦ O *kernel* implementa a funcionalidade mínima referente a
 - gestão básica da *CPU*
 - gestão de memória
 - suporte de *I/O*
 - comunicação entre processos
- ♦ A restante funcionalidade do S.O. é implementada em *proc.^{os}* de sistema que correm em modo de utilizador; estes processos comunicam entre si através de mensagens (modelo cliente-servidor)

Primeiro sistema baseado em *microkernel*: Hydra (CMU, 1970)
Outros exemplos: Mach (CMU), Chorus (Unix-like, francês), Minix

Windows NT - estrutura *microkernel* modificada;
ao contrário de uma arquitectura *microkernel* "pura"
muitas das funções de sistema fora do *microkernel*
executam em modo *kernel*, por razões de performance

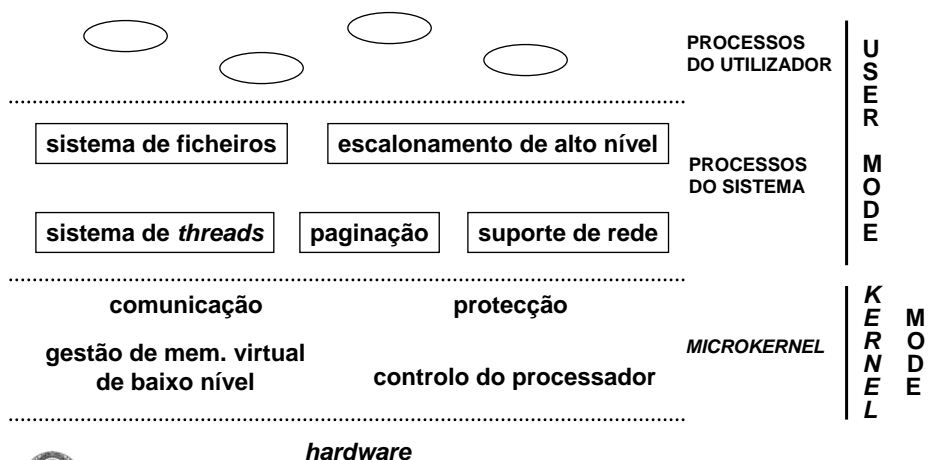


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Estrutura baseada em *microkernel*:

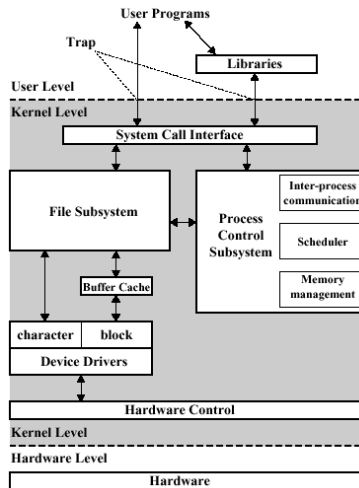


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Arquitectura do Unix (Bach)

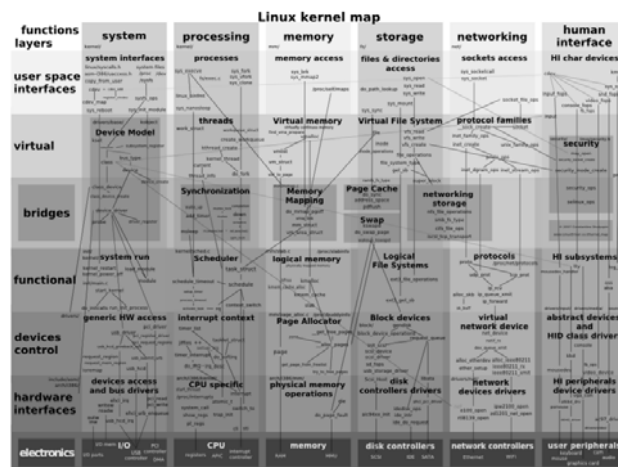


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Arquitectura do Linux

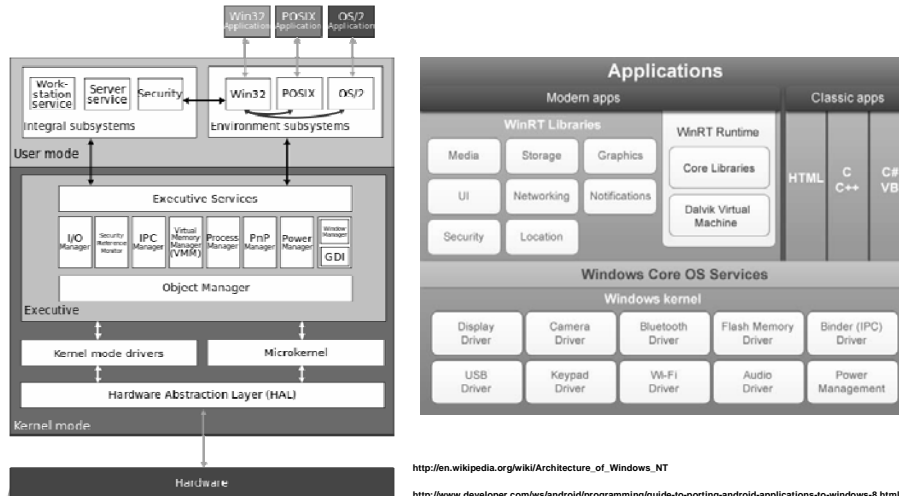


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Arquitectura do Windows NT & Windows 8

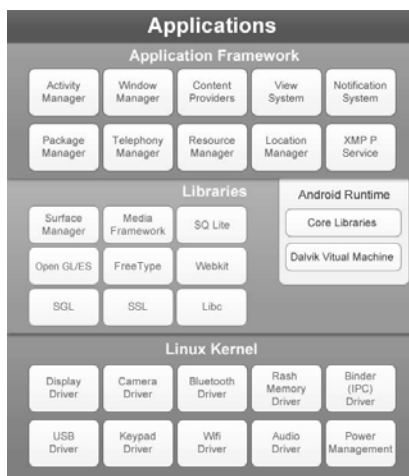


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

Arquitectura do Android



<http://www.developer.com/ws/android/programming/guide-to-porting-android-applications-to-windows-8.html>



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

A seguir: tópicos sobre S.O.'s

