Chp. 2 - Operating System Overview

2.1 - Objetivos e Funções de um Sist. Operacional

- **tópicos:** sist. oper. e a interface usuário/computador; sist. oper. como um gerenciador de recursos; evolução/melhoramento de um sist. operacional.

2.2 - Evolução dos Sistemas Operacionais

- **tópicos:** sist. com processamento serial sist. em lote; sist. em lote multiprogramado; sist. de tempo compartilhado.

2.3 - Principais Funcionalidades e/ou Requisitos do S.O.

2.4 - Características dos Sistemas Operacionais Modernos

2.5 - Visão Geral do Windows NT

2.6 - Sistema UNIX Tradicional

2.7 - Sistema UNIX Moderno

... Chp. 2 - Referências Bibliográficas

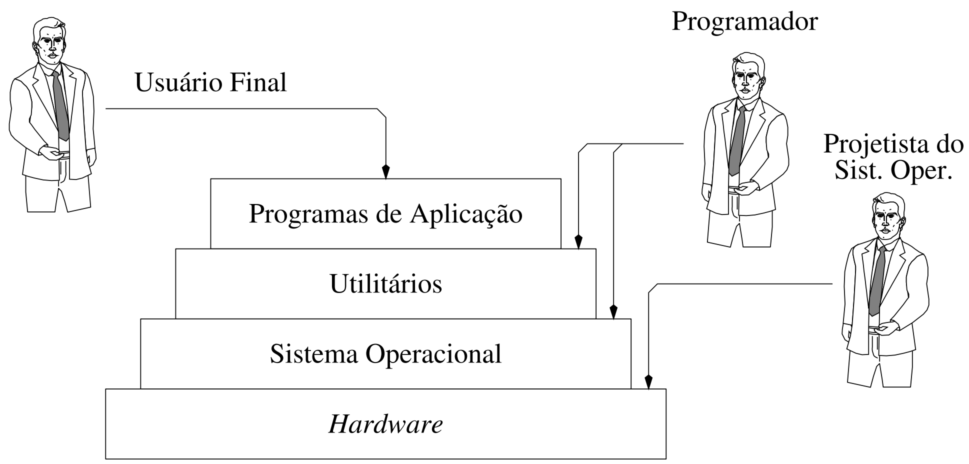
✯ William STALLINGS; **Operating Systems: Internals and Design Principles**, New Jerssey, Prentice-Hall, 1998, ISBN: 0-13-887407-7

✯ Eleri CARDOZO; Maurício MAGALHÃES; Luís F. FAINA; **Sistemas Operacionais**, Dep. de Eng. de Computação e Automação Industrial, Fac. de Engenharia Elétrica e de Computação, UNICAMP, 1996.

2.1 - Objetivos e Funções de um Sist. Operacional

- **definição:** sistema operacional é um programa que controla a execução de programas de aplicação e age como uma interface entre o usuário do computador e o *hardware* do computador.

- **objetivos:** conveniência, eficiência e habilidade para evoluir e/ou ser atualizado.



2.1.1 - Sist. Oper. e a Interface Usuário/Computador

- **criação de programas** - o sist. oper. provê uma variedade de funcionalidades e serviços para assistir o programador na criação de programas;

- **execução de programas** - ... instruções e dados precisam ser carregados na memória, dispositivos de entrada/saída e arquivos precisam ser inicializados e outros recursos devem ser preparados;

- **dispositivos de entrada/saída** - como cada dispositivo de entrada/saída apresenta suas próprias peculiaridades, tais como, conjunto de instruções e sinais de controle para operação, cabe ao sist. oper. abstrair estes detalhes para o usuário;

- **acesso restrito a arquivos** - cabe ao sist. oper. esconder do usuário a natureza do dispositivo de armazenamento, bem como o formato do arquivo armazenado;

- **acesso ao sistema** - funções de acesso devem proteger recursos e dados de usuários não autorizados, bem como resolver conflitos de disputa por recursos;

... 2.1.1 - Sist. Oper. e a Interface Usuário/Computador

- **detecção de erros** - uma variedade de erros podem ocorrer: erros de *hardware* externos e internos, erros de *software* e erros decorrentes da inabilidade do sist. oper. em garantir uma requisição da aplicação do usuário;

- **resposta aos erros** - em cada caso, o sist. oper. pode: relatar o erro ao usuário, repetir a operação ou abortar a aplicação;

- **coleta de medidas** - um bom sist. oper. deve coletar estatísticas acerca de vários recursos bem como monitorar parâmetros de performance (p.ex.: tempo de resposta):
  - em um dado sistema, estas informações podem ser usadas para antecipar a necessidade de melhorias ou ajustes no sistema;
  - em sistemas multi-usuários, estas informações podem ser usadas para propósitos de tarifação.

2.1.2 - Sist. Oper. como um Gerenciador de Recursos

✰ **função:** é responsável pelo gerenciamento dos recursos computacionais responsáveis pela transferência, armazenamento e processamento de dados bem como para controle dessas funções.

- ... como mecanismo de controle, cabe duas ressalvas:

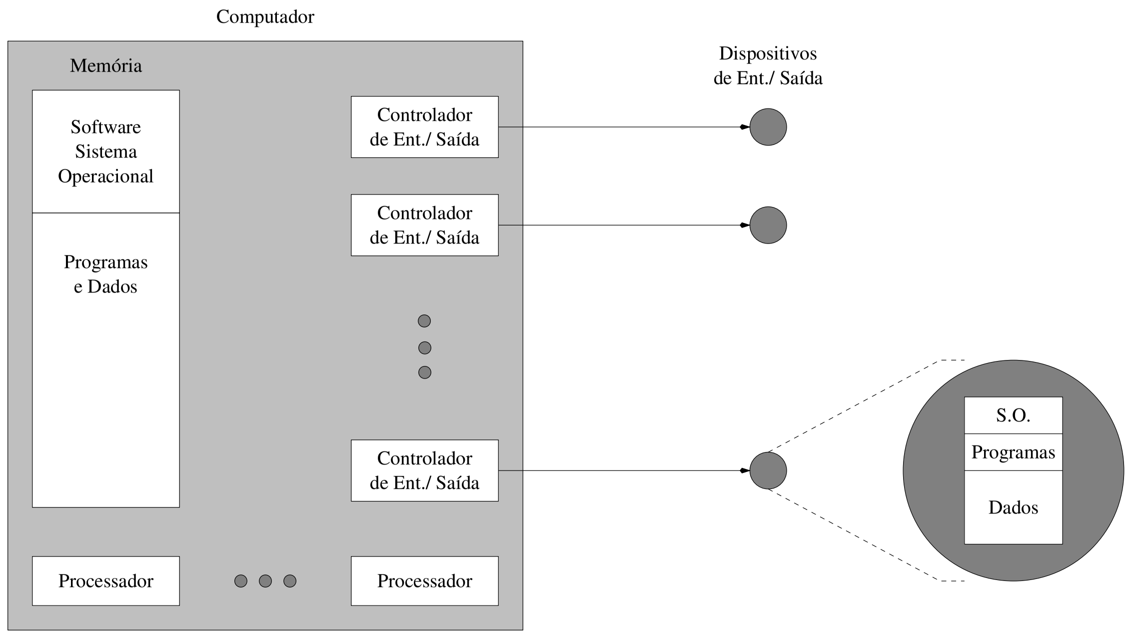
  ❶ um sistema operacional funciona da mesma que um programa comum em um computador, ou seja, as funções do sistema operacional são pequenos códigos de programas executados pelo processador;

  ❷ freqüentemente o sistema operacional renuncia o controle do processador e depende do processador para readquirí-lo novamente.

- O Sistema Operacional não é nada mais que um Programa de Computador.

... 2.1.2 - Sist. Oper. como um Gerenciador de Recursos

- ... a figura abaixo sugere os principais elementos gerenciados pelo sistema operacional:



2.1.3 - Evolução/Melhoramentos de um Sistema Operacional

✡ Os Sist. Operacionais evoluem ao passar dos anos por uma série de razões:

❶ atualização do *hardware* e/ou novos *hardwares* - diferentes funcionalidades requerem suporte diferenciado no sistema operacional;
❷ novos serviços - em resposta às necessidade dos usuários e/ou gerentes de sistemas, um sistema operacional se expande para oferecer novos serviços;
❸ correção de *bugs* - falhas são descobertas ao longo do tempo e devem ser reparadas, o que também constitui numa fonte de falhas.

- ... a necessidade de mudanças em um sistema operacional exige que certos requisitos sejam considerados no seu projeto (p. ex., modularidade no seu desenvolvimento).

- ... por outro lado, para programas mais complexos (maior porte) uma modularização simples provavelmente será inadequada, ou seja, necessitaremos muito mais que um simples particionamento do programa em sub-rotinas (p. ex., orientação a objetos).

2.2 - A Evolução dos Sistemas Operacionais

✰ De modo a atender os requisitos chaves de um sistema operacional e a importância dos principais aspectos nos sistemas operacionais contemporâneos, é usual analisar como se deu a evolução dos sistemas operacionais.

✰ **2.2.1 - Serial Processsing:** ... nos primeiros computadores (1940 até meados de 1950), o programador interagia diretamente com o *hardware*, ou seja, não havia sistema operacional para auxiliar o programador.

- as máquinas eram executadas a partir de *consoles*, constituídos de lâmpadas sinalizadoras, chaves, algum dispositivo de entrada e impressora;
- se um erro ocorresse durante a execução do programa, a indicação se dava através das lâmpadas sinalizadoras (*leds*) — com essas informações o programador examinava os registros para determinar a causa do erro;
- se a execução do programa evoluia como esperado, os dados gerados como resposta a sua execução eram descarregados na impressora.

... 2.2.1 - Serial Processsing (Evolução dos Sist. Oper.)

✘ **escalonamento** - a reserva da máquina se dava por um período de tempo múltiplo de um intervalo preestabelecido (p. ex., 30 minutos) o que não permitia na maioria das vezes o melhor uso do sistema computacional;

✘ **tempo de preparação** - a execução de um simples programa (isto é, *job*) involvia carregar o compilador e o programa na memória, gerar o programa objeto bem como ligá-lo as funções comuns, e cada um desses passos comumente exigia a conexão com dispositivos como *tapes*, leitora de cartões, etc.

- na ocorrência de erros, todo o processo precisava ser reiniciado;
- ... muito tempo era perdido na preparação do ambiente para que só depois o programa do usuário fosse executado.

✰ Este modo de operação denominado *Serial Processing* reflete o fato de usuários acessarem o computador em seqüência — para torná-lo mais eficiente, programas utilitários foram desenvolvidos (p.ex., bibliotecas de funções comuns, *linkers*, carregadores, *debuggers*, *drivers* de entrada/saída).

2.2.2 - Batch Systems (Evolução dos Sist. Oper.)

✰ As máquinas, até então, eram muito caras e apresentavam tempo para escalonamento e preparação inaceitáveis e, portanto, a palavra de ordem era maximizar a utilização das máquinas.

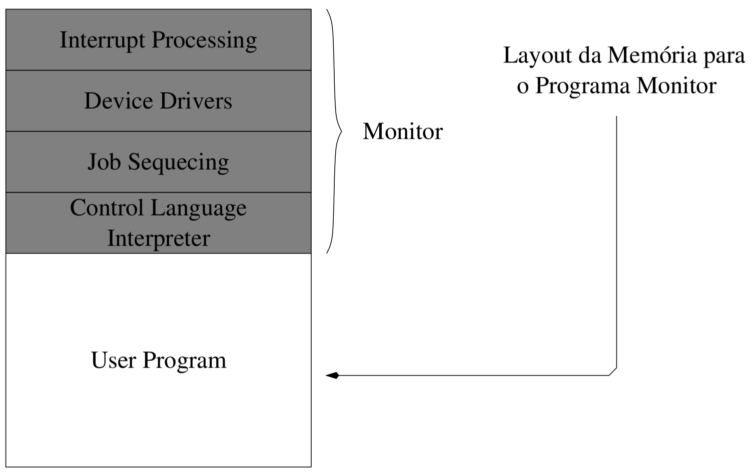
- ... para melhorar a utilização das máquinas, o conceito de Sist. Operacional em Lote foi desenvolvido em meados de 1950 pela General Motors para uso no IBM 701.
- ... refinamentos vieram a *posteriori* e por volta de 1960 mais de um fornecedor já havia desenvolvido sist. oper. em lote para os seus computadores — p.ex.: o IBSYS (Sist. Oper. para o IBM 7090/7094 influenciou largamente outros sistemas).

✰ **idéia** - utilização de um programa denominado **monitor** responsável pelo gerenciamento de cada programa do lote (*batch system*).

  - o monitor permanece na memória principal para execução a qualquer instante e seus utilitários são carregados quando necessário;

... 2.2.2 - Batch Systems (Evolução dos Sist. Oper.)

- ... o monitor lê um *job* de cada vez do dispositivo de entrada;
- ... o monitor carrega o *job* para a área de programas do usuário;
- ... um instrução do monitor gera um desvio para o começo do prog. do usuário;
- ... a execução do programa continua até terminar ou ocorrer um erro.



... 2.2.2 - Batch Systems (Evolução dos Sist. Oper.)

✰ **Monitor** ou **Sist. Oper. em Lote** é simplesmente um programa de computador — ele repousa na habilidade do processador buscar instruções de várias partes da memória de maneira a tomar ou liberar alternadamente o controle.

✰ Algumas considerações desejáveis nestes Sistemas:

- **proteção da memória** - enquanto um programa está executando, este não tem autorização para alterar a área de memória que contém o monitor;
- **temporizador** - um temporizado é utilizado para prevenir que um simples *job* monopolize o sistema;
- **instruções com privilégio** - algumas instruções são projetadas para somente serem executadas pelo monitor (p. ex., instruções de entrada/saída);
- **interrupções** - concedem maior flexibilidade ao sistema operacional quando da renúncia ou readmissão do controle para programas do usuário.

✰ Naturalmente que um S.O. não precisa ser desenvolvido levando em consideração esses aspectos, mas desenvolvedores de sistemas aprenderam que tal atitude é catastrófica — então, até os Batch Systems foram contemplados com essas considerações.

2.2.3 - Multiprogrammed Batch Systems (Evolução dos Sist. Oper.)

✘ operações de entrada/saída são excessivamente lentas comparadas com a execução de instruções convencionais de um programa;

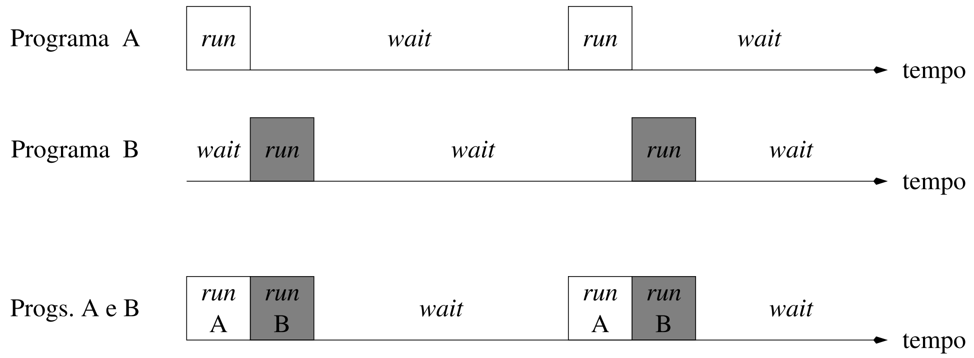
✘ um programa mesmo contendo um número pequeno de operações de entrada/saída, gastaria a maior parte de seu tempo esperando por elas;

✘ temos assim, um pobre uso da CPU quando somente um programa esta presente na memória — o processador continua livre e, portanto, pode ser utilizado.

✰ **sugestão:** expandir a memória para suportar 2, 3 ou mais programas bem como permitir o chaveamento da execução entre os mesmos — este processo é conhecido como **multiprogramação** ou **multitarefa**

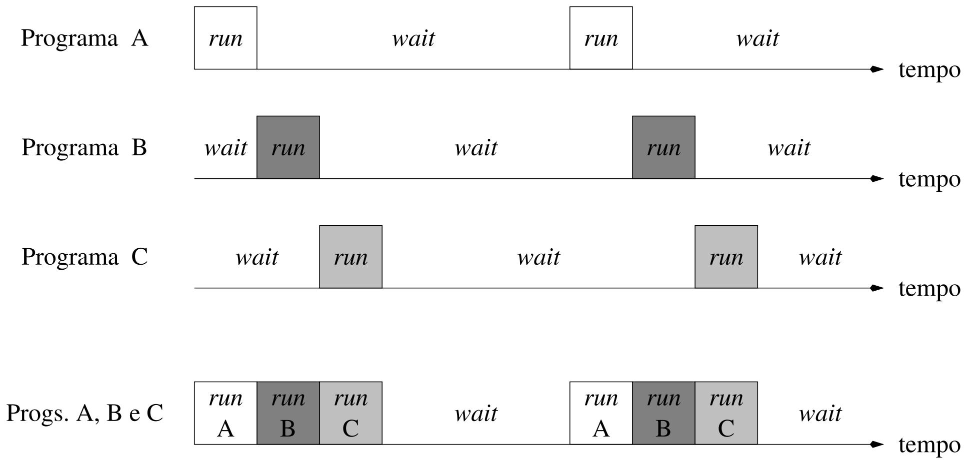
... 2.2.3 - Multiprogrammed Batch Systems

- ... vejamos alguns exemplos de multiprogramação com 02 processos;



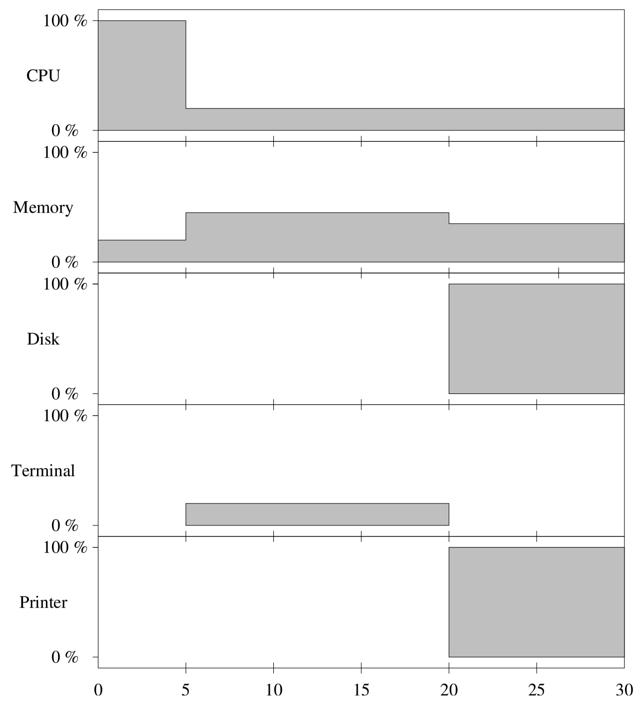
... 2.2.3 - Multiprogrammed Batch Systems

- ... vejamos alguns exemplos de multiprogramação com 03 processos;



... 2.2.3 - Multiprogrammed Batch Systems

- ... para ilustrar os benefícios da multiprogramação, considere:



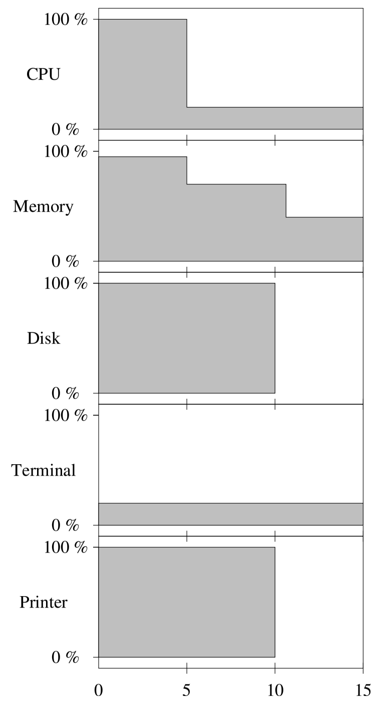
Characteristics of the Computer:

Memory: 256 K words
I/O Modules: Disk, Terminal and Printer

	Job #1	Job #2	Job #3
Type of Job	Heavy	Heavy I/O	Heavy I/O
Duration	5 min.	15 min.	10 min.
Memory required	50 K	100 K	80 K
Need Disk?	No	No	Yes
Need Terminal?	No	Yes	No
Need Printer?	No	No	Yes

... 2.2.3 - Multiprogrammed Batch Systems

- ... para ilustrar os benefícios da multiprogramação, considere:



Characteristics of the Computer:

Memory: 256 K words
I/O Modules: Disk, Terminal and Printer

	Uniprogramming	Multiprogramming
Processor Time	17 %	33 %
Memory Use	30 %	67 %
Disk Use	33 %	67 %
Printer Use	33 %	67 %
Elapsed Time	30 min.	30 min.
Throughput Rate	6 jobs/h	6 jobs/h
Mean Response Time	18 min.	18 min.

... 2.2.3 - Multiprogrammed Batch Systems

✯ Assim com nos *Batch Systems*, *Multiprogrammed Batch Systems* devem assumir algumas considerações de *hardware* e *software*:

- **interrupções de entrada/saída e DMA** - de forma a executar instruções enquanto dispositivos de entrada/saída estão ocupados;
- **gerenciamento de memória** - diversos *jobs* prontos para serem executados devem ser mantidos em memória;
- **proteção da memória** - dados e programa;
- **escalonamento** - com vários processos prontos para serem executados é necessário decidir que deve ser escalonado;
- **gerenciar a disputa por recursos** - não só no caso anterior, como para os demais recursos de entrada/saída.

2.2.4 - Time-Sharing Systems (Evolução dos Sist. Oper.)

✘ Com o uso da Multiprogramação, *Batch Systems* podem ser eficientes, entretanto, para um número grande de *jobs* é desejável que o usuário possa interagir diretamente com o computador (ou seja, modo interativo).

✯ Entretanto, assim como na Multiprogramação, o processador pode ser utilizado para manipular *jobs* interativos — neste caso, a técnica é referenciada como *Time-Sharing* uma vez que o tempo do processador é partilhado entre os usuários.

- ... em Sist. *Time-Sharing*, múltiplos usuários acessam simultaneamente o sistema através de terminais, com o S.O. intercalando a execução de cada programa do usuário em pequenas porções (*quantum*) de tempo de uso do recurso computacional.

	Multiprogrammed Batch Systems	Time-Sharing Systems
Objetivos Principais	maximizar o uso do processador	'minimizar' o tempo de resposta
Fonte de Instruções para o Sist. Oper.	instruções de linguagem de controle do job presentes no job	comandos informados no terminal pelo usuário

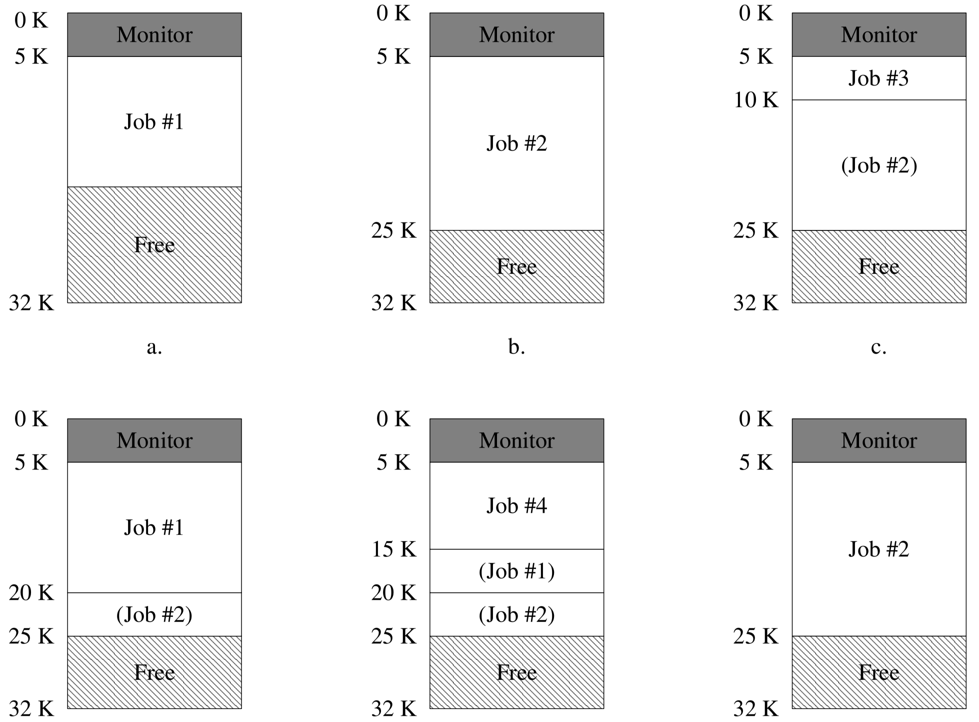
... 2.2.4 - Time-Sharing Systems (Evolução dos Sist. Oper.)

✣ 1º S.O. *Time-Sharing - Compatible Time-Sharing System* foi desenvolvido no MIT no Projeto MAC (*Machine-Aided Cognition* ou *Multiple-Access Computers*).

✣ O CTSS foi inicialmente desenvolvido em 1961 para o IBM 709 e mais tarde transferido para o IBM 7094 — algumas de suas peculiaridades:

- o sistema executava em um máquina com 32 K palavras de 36 bits na memória principal, com um monitor residente consumindo 5 K palavras;
- programas e dados do usuário eram carregados nos 27 K restantes da memória;
- o *clock* gerava interrupções a uma taxa de 0.2 segundos;
- a cada interrupção de relógio, o sist. operacional readquiria o controle para atribuí-lo a outro processo de um outro usuário;
- na troca de contexto, os dados e programa do usuário eram gravados em disco para que o novo programa e dados pudessem ser carregados.
- para minimizar o tráfego em disco, a memória do usuário era sobreescrita quando havia um programa para ser executado.

... 2.2.4 - Time-Sharing Systems (Evolução dos Sist. Oper.)



a. b. c. d. e. f.

2.3 - Principais Funcionalidades e/ou Requisitos

✰ Sistemas Operacionais estão entre os mais complexos *softwares* já desenvolvidos e, neste sentido, refletem o desafio de convergência entre diferentes objetivos e até mesmo competitivos tais como: conveniência, eficiência e habilidade para evoluir.

- ... para o projeto e desenvolvimento de sistemas operacionais, 05 funcionalidades e/ou realizações devem ser consideradas:

  ❶ criação e controle de processos;
  ❷ gerenciamento de memória;
  ❸ segurança e proteção de informação;
  ❹ escalonamento e gerenciamento de recursos;
  ❺ estrutura no sistema operacional.

- ... cada funcionalidade é caracterizada por princípios ou abstrações, desenvolvidas para solucionar problemas de ordem prática.

- ... tomados juntos, esses 05 pontos constituem os aspectos chaves de projeto e implementação dos sistemas operacionais contemporâneos.

2.3.1 - Criação e Controle de Processos

- ... o conceito de **processo** é fundamental na estrutura dos sistemas operacionais e muitas definições tem sido atribuídas a este termo:

  ❶ um programa em execução;
  ❷ o espírito animado (*animated spirit*) de um programa;
  ❸ a entidade que pode ser designada para ser executada por um processador.

✰ **2.3.2 - Gerenciamento de Memória** - usuários necessitam de um ambiente que suporte programação modular e o uso flexível de dados, enquanto gerenciadores de sistemas necessitam controle ordenado e eficiente para alocação de memória.

  ❶ não interferência entre os processos;
  ❷ gerenciamento e alocação automática;
  ❸ suporte à programação modular;
  ❹ segurança e controle de acesso;
  ❺ armazenagem da informação por longos períodos.

2.3.3 - Segurança e Proteção da Informação

- ... muito do trabalho de proteção e segurança do sistema operacional pode ser enquadrado (agrupado) em 03 categorias:

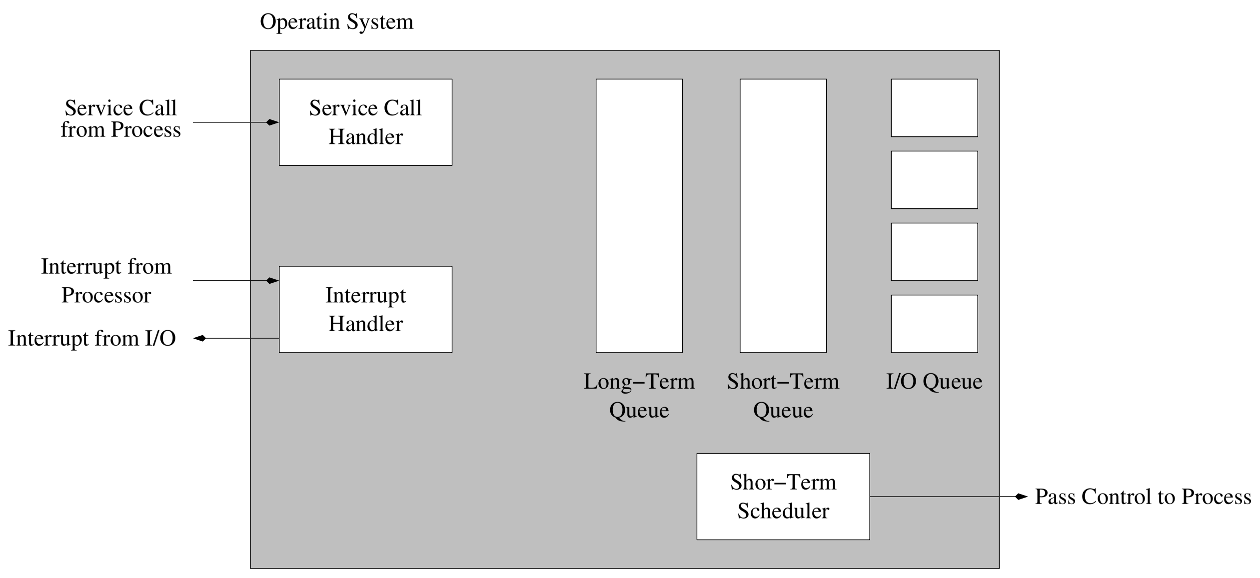
  ❶ controle de acesso - em relação aos usuários e processos;
  ❷ controle do fluxo de informação - regulamenta o fluxo de dados dentro do sistema bem como sua entrega aos usuários aos quais se destina;
  ❸ certificação - garantir que os mecanismos anteriores descritos sejam contemplados segundo políticas de segurança e proteção.

✰ **2.3.4 - Escalonamento e Gerenciamento de Recursos** - 03 fatores devem ser considerados na alocação e políticas de escalonamento de recursos:

  ❶ eqüidade na alocação de recursos;
  ❷ tratamento diferenciado p/ diferentes classes de processos;
  ❸ eficiência - maximizar a vazão e minimizar o tempo de resposta dos processos.

... 2.3.4 - Escalonamento e Gerenciamento de Recursos

- ... a figura abaixo sugere os principais elementos de um sistema operacional:



2.3.5 - Estrutura do Sistema Operacional

✯ Mais e mais considerações (aspectos de projeto) têm sido adicionados aos sistemas operacionais e, com o *hardware* tem se tornado mais complexo e versátil, o tamanho e complexidade dos sistemas operacionais têm crescido.

- ... problemas mais comuns dos sistemas operacionais contemporâneos:

❶ estão em parte "desatualizados" quando entregues;
❷ latência dos *bugs* nos sistemas;
❸ a performance não é o que se esperava.

- ... requisitos na estrutura de *software* dos sistemas operacionais contemporâneos:

❶ programação modular - auxilia a organização do processo de desenvolvimento bem como minimiza a tarefa de diagnóstico e retirada dos erros;
❷ abstração da informação em níveis hierárquicos - a estrutura hierárquica do sistema operacional separa suas funções de acordo com sua complexidade, com suas característica na escala de tempo e com seus níveis de abstração;

... 2.3.5 - Estrutura do Sistema Operacional

✯ Idealmente, os níveis devem ser definidos de forma que as mudanças em um nível não requeiram mudanças em outros níveis — deste modo, teremos decomposto um problema em um número maior de subproblemas gerenciáveis.

- **lower level** - interagem diretamente com o *hardware* do computador;
- **upper level** - interagem diretamente com o usuário do sistema.

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User Processes	User Processes	e.g., quit, kill, suspend, resume
11	Directories	Directories	e.g., create, destroy, attach, detach, search, list
10	Devices	External Devices (printers, displays)	e.g., create, open, close, read, write
9	File System	Files	e.g., create, destroy, open, close, read, write
8	Communications	Pipes	e.g., create, destroy, open, close, read, write
7	Virtual Memory	Segments, Pages	e.g., read, write, fetch
6	Local Secondary Store	Blocks of Data and Device Channels	e.g., read, write, allocate, free
5	Primitive Processes	Primitive Processes, Semaphores, Ready List	e.g., suspend, resume, wait, signal
4	Interrupts	Interrupt handling programs	e.g., invokes, mask, unmask, retry
3	Procedures	Procedures, call stack, display	e.g., mark stack, call, return
2	Instruction Set	Evaluation stack, microprogram interpreter,	e.g., load, store, add, subtract, branch
1	Eletrocnic Circuits	Registers, Gates, Buses, etc.	e.g., clear, transfer, activate, complement.

2.4 - Características dos Sistemas Operacionais Modernos

✠ Uma grande variedade de diferentes abordagens e elementos de projeto foram introduzidos em Sistemas Operacionais experimentais e comerciais:

- ... em resposta aos novos desenvolvimentos de *hardware*: máquinas multiprocessadas; redes de alta-velocidade; variedade de dispositivos de armazenamento.

- ... em resposta aos novos desenvolvimentos de *software*: aplicações multimídia; acesso na Web e Internet; paradigma cliente/servidor.

- ... uma grande variedade de abordagens e elementos de projeto têm testadas nos sistemas operacionais comerciais e experimentais, mas muito deste trabalho pode ser categorizado nas seguintes categorias:

  - ✪ Arquitetura *Microkernel*;
  - ✪ Multiprocessamento Simétrico;
  - ✪ Projeto Orientado a Objetos.
  - ✪ *Multithreading*;
  - ✪ Sistemas Operacionais Distribuídos;

... 2.4 - Características dos Sistemas Operacionais Modernos

✠ **microkernel architecture:** acomoda somente um pequeno conjunto de funções essenciais no *kernel*, incluindo espaço de endereçamento, comunicação entre processos e escalonamento básico;

- ... esta abordagem desacopla o desenvolvimento do *kernel* do desenvolvimento de servidores de aplicações e/ou funcionalidades;

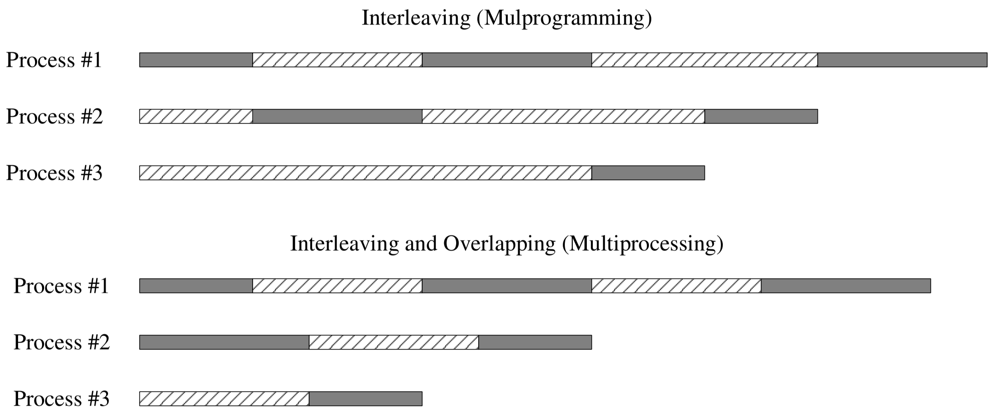
- ... o projeto orientado a *microkernel* simplifica a implementação, oferece flexibilidade e se ajusta prontamente às características dos sistemas distribuídos.

✠ **multithreading:** é uma técnica que divide o processo em unidades escalonáveis que podem ser executadas simultaneamente;

- ... útil em aplicações que acomodam tarefas essencialmente independentes e que não necessitam serem serializadas, p.ex.: servidor de base de dados que aceita simultaneamente inúmeras requisições de processos clientes.

... 2.4 - Características dos Sistemas Operacionais Modernos

✰ *Symmetric-Multiprocessing* oferece inúmeras vantagens, dentre elas: performance, disponibilidade, crescimento incremental e escalabilidade.



... 2.4 - Características dos Sistemas Operacionais Modernos

✰ **distributed operating system:** provê a ilusão de um único espaço de endereçamento de memória principal bem como espaço de endereçamento secundário e funcionalidades adicionais como um sistema de arquivo distribuído;

- ... a idéia é oferecer a aparência de um único sistema para um *cluster* de computadores, ou seja, um sistema com múltiplos computadores.

✰ **object-oriented design:** além de oferecer disciplina na adição de extensões de módulos a *kernels*, também facilita o desenvolvimento de aplicações distribuídas.

2.5 - Visão Geral do Windows NT

✰ **Windows NT 3.1:** liberada em 1983, mantinha a mesma interface do Windows 3.1, mas já era um sistema operacional de 32 bits com a habilidade de suportar o DOS e Aplicações Windows bem como suporte para o OS/2;

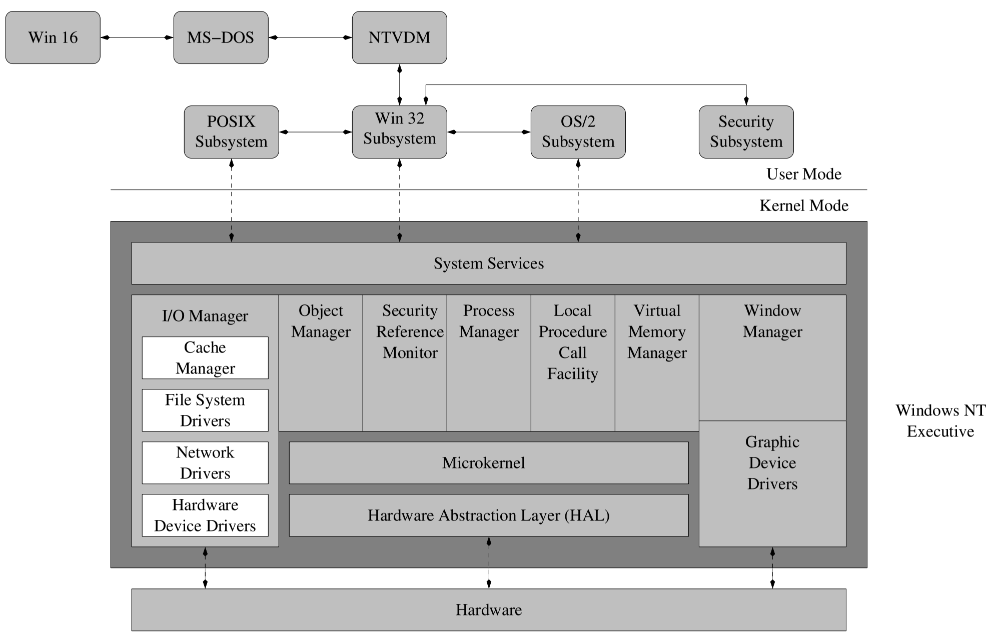
- ... após várias versões do NT 3.x, o NT 4.0 apresentava a mesma arquitetura interna como nas versões 3.x, entretanto, vários componentes gráficos passaram a ser executados em modo *kernel* (*Windows NT Executive*);

- ... em 1998, a Microsoft lançou o NT 5.0, que novamente mantinha a mesma arquitetura *microkernel* do NT 4.0 e acrescentava funções e serviços para aplicações distribuídas, p.ex., *Active Directory* - diretório distribuído;

- ... a distinção entre NT Server e NT Worksation é que o NT Server incluia alguns serviços essenciais para que o NT funcionasse como um Servidor de Rede.

... 2.5 - Visão Geral do Windows NT

✰ A estrutura geral do Windows NT é altamente modular conferindo-lhe alta flexibilidade podendo ser executado em diferentes plataformas;



... 2.5 - Visão Geral do Windows NT

✰ **NT Executive:** altamente modular, acomoda alguns princípios da arquitetura *microkernel*, pois isso, chamada *modified microkernel architecture*;

- ... a razão é que acomoda funções de sistema projetadas para rodar no modo usuário, mas que de fato são executadas no modo *kernel* por razões de performance;

- ... um dos aspectos de projeto é a sua portabilidade — capacidade de ser executado não somente sobre uma plataforma Intel mas em uma variedade de plataformas;

✰ Para satisfazer este aspecto de projeto, o *NT Executive* enxerga a mesma infra-estrutura de *hardware* através de sua estrutura em camadas:

- **Hardware Abstraction Layer**
- **Microkernel**;
- **Executive Services**;
- **System Services.**

... 2.5 - Visão Geral do Windows NT

✰ **Hardware Abstraction Layer:** mapeia comandos e respostas do *hardware* genérico para aqueles específicos e dependentes da plataforma, p.ex.: Arquitetura Pentium, Power PC e Alpha;

- ... é de sua responsabilidade o barramento do sistema, o controlador do acesso direto a memória, temporizadores do sistema, visão do módulo de memória pelo *kernel* e até suporte necessário ao multiprocessamento simétrico.

✰ **Microkernel:** responsável pelo gerenciamento do escalonamento de *threads*, chaveamento de processos, manipulação de interrupções e exceções e sincronização de multiprocessadores;

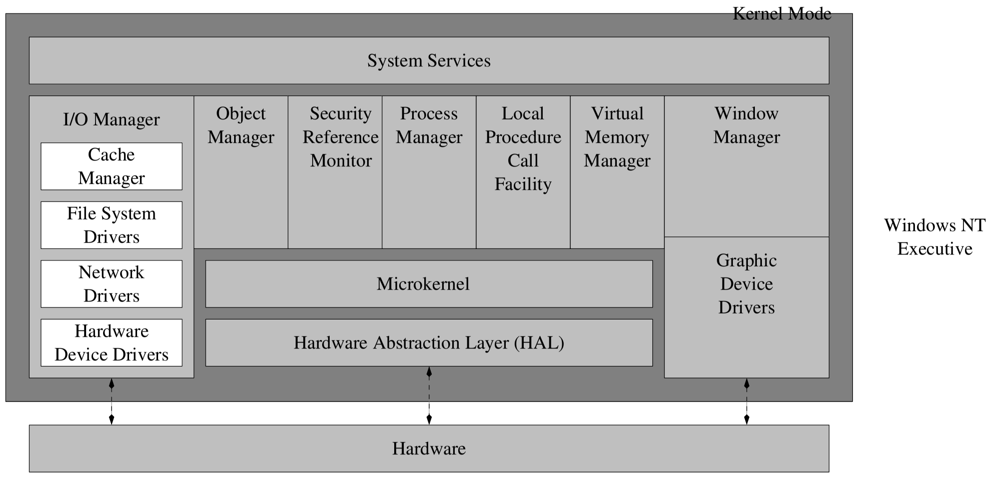
- ... única parte do sistema operacional que não é preemptável ou sujeita a paginação, ou seja, constitui um componente fundamental e muito usado pelo sistema.

... 2.5 - Visão Geral do Windows NT

✰ **Executive Services:** inclui uma variedade de módulos para funções específicas do sistema, todas acessando o *hardware* através do HAL;

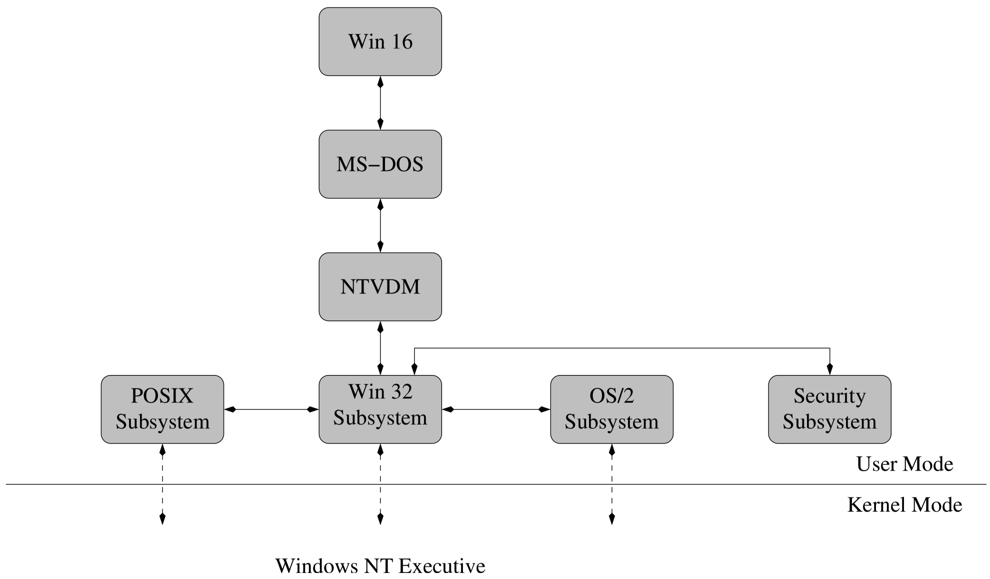
- ... as duas únicas exceções são: *Window/Graphic* e o *I/O Manager*.

✰ **System Services:** provê uma interface para *software* no modo usuário.



... 2.5 - Visão Geral do Windows NT

✰ NT é estruturado para suportar aplicações escritas para o Windows NT, Windows 95, e vários outros Sistemas Operacionais, p.ex.: OS/2 e MS-DOS.



... 2.5 - Visão Geral do Windows NT

✰ A maneira pela qual o Windows NT oferece este suporte é através de subsistemas protegidos de ambiente (*environment subsystems*);

- ... um subsistema protegido disponibiliza uma interface gráfica ou de comando de linha que define o aspecto e funcionamento do sistema operacional para o usuário;

✰ São 05, os subsistemas suportados pelo Windows NT:

  - MS-DOS NT Virtual DOS Machine (NTVDM);
  - Windows 16 NTVDM;
  - OS/2 Subsystem;
  - POSIX Subsystem;
  - Windows 32 Subsystem.

... 2.5 - Visão Geral do Windows NT

✰ No NT todos os subsistemas e aplicações estão estruturados através do modelo computacional Cliente/Servidor, bastante comum na computação distribuída;

- ... cada subsistema de ambiente e o subsistema de serviço executivo são implementados com um ou mais processos, cada qual espera por uma requisição de um cliente para o serviço que oferece, p.ex.: serviço de instanciação de processos, etc.

- ... um cliente que pode ser uma programa de aplicação ou um outro módulo de sistema operacional requisita um serviço enviando um mensagem;

- ... a mensagem é roteada através do subsistema executivo para o servidor apropriado, que realiza a operação requisitada e retorna o resultado ou informação de *status* através de uma outra mensagem (também roteada pelo subsistema executivo).

... 2.5 - Visão Geral do Windows NT

✰ Modelo Cliente/Servidor proporciona muitas vantagens:

- **simplificação do NT executivo:** possibilita o desenvolvimento de uma variedade de APIs sem conflitos e duplicações no subsistema;
- **aumento da confiabilidade:** cada módulo executivo é executado em um processo separado, com sua própria partição de memória protegida de outros módulos;
- **interface uniforme de comunicação:** o processo de troca de mensagens é escondido da aplicação cliente por meio dos *stubs* mantidos em bibliotecas dinâmicas;
- **infra-estrutura básica para sistemas distribuídos:** tipicamente, sistemas distribuídos utilizam-se do modelo cliente/servidor com chamada de procedimento remoto implementada usando módulos servidores e clientes distribuídos.

... 2.5 - Visão Geral do Windows NT

✰ Finalmente, Windows NT segue a abordagem de orientação a objetos, o que traz facilidades no compartilhamento de recursos e dados entre processos bem como proteção de recursos contra acessos não autorizados;

- **encapsulação:** um objeto consiste de um ou mais atributos e, um ou mais procedimentos denominados de serviços ou métodos;
- **classes e instâncias:** uma classe de objeto é um *template* que lista os atributos e serviços de um objeto e define certas características dos objetos;
- **herança:** embora não seja suportado no nível do usuário, é suportado em certo nível pelo subsistema executivo, p.ex.: objeto diretório como um objeto container;
- **poliformismo:** utilizado internamente para manipular objetos do tipo *any*, entretanto o NT não contempla esta propriedade totalmente, dado que muitas APIs são específicas para tipos específicos de objetos.

2.6 - Sistema UNIX Tradicional

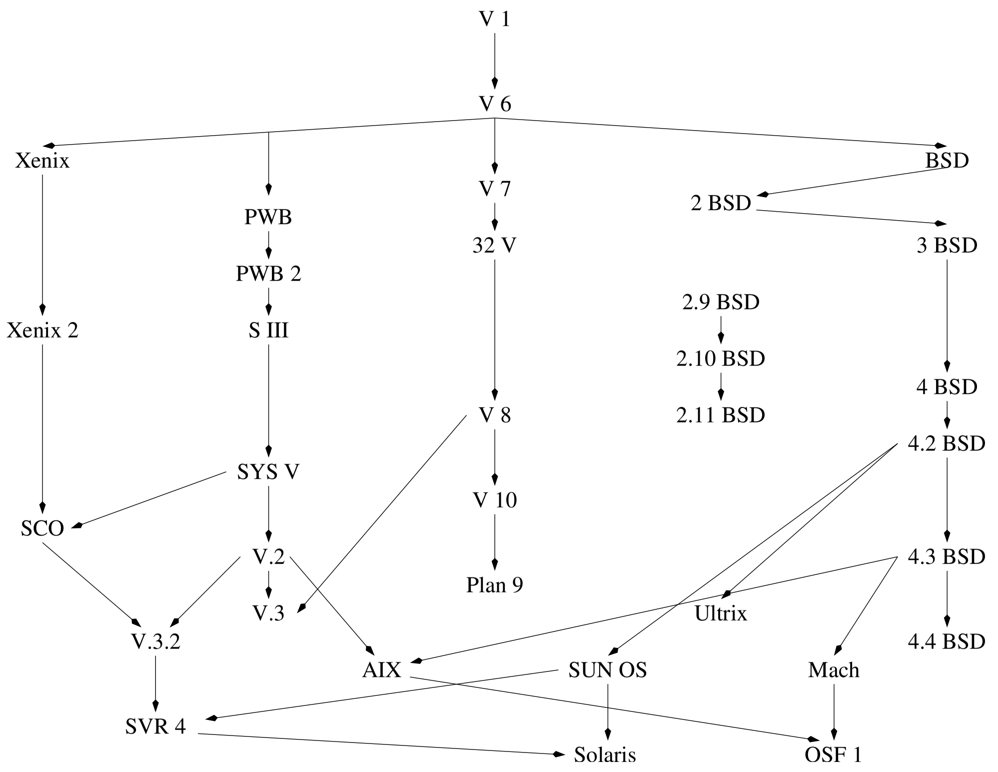
✰ UNIX foi inicialmente desenvolvido pelo Bell Labs e tornou-se operacional no PDP-7 em 1970, que parcialmente teve o apoio de pessoas que participaram do Projeto MAC do MIT, no qual foi desenvolvido o CTSS e depois o Multics;

- … após vários melhoramentos e desenvolvimentos para novas plataformas, o UNIX foi reescrito em Linguagem C algo inconcebível na época, posto o sentimento era de que algo complexo como o sistema operacional deveria ser escrito em *assembly*;

- … a reescrita do código em C demonstrou as vantagens de se usar uma linguagem de alto-nível de abstração para a maior parte do código senão todo e, atualmente, todas as implementações UNIX são escritas em C e C++;

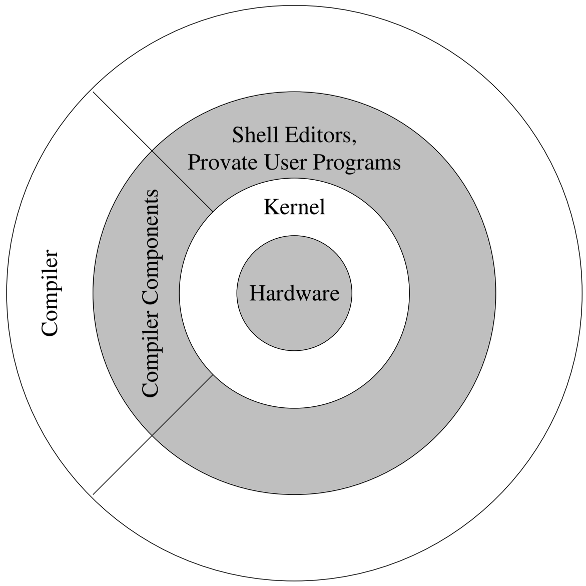
- … a primeira versão disponibilizada fora do Bell Labs foi a Versão 6 em 1976, seguida pela Versão 7 em 1978, a antecessora dos Sistemas UNIX Modernos.

… 2.6 - Sistema UNIX Tradicional



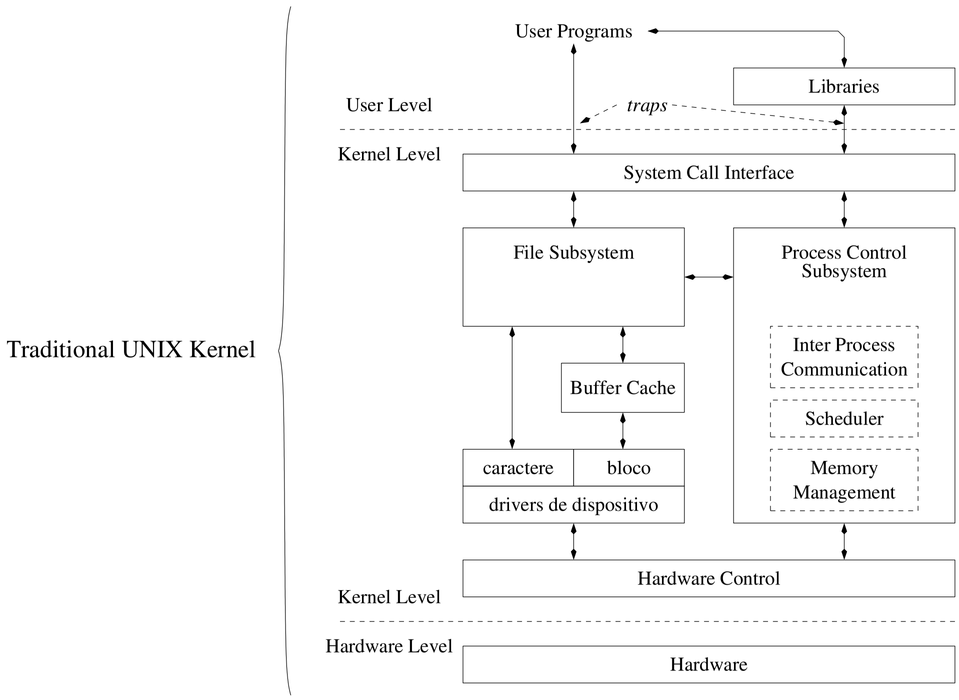
... 2.6 - Sistema UNIX Tradicional

✰ **kernel:** isolado do usuário e das aplicações, abstrai todas as funcionalidades do *hardware* intermediando todas as requisições para ele.



... 2.6 - Sistema UNIX Tradicional

✰ Visão detalhada do kernel do Sistema UNIX:

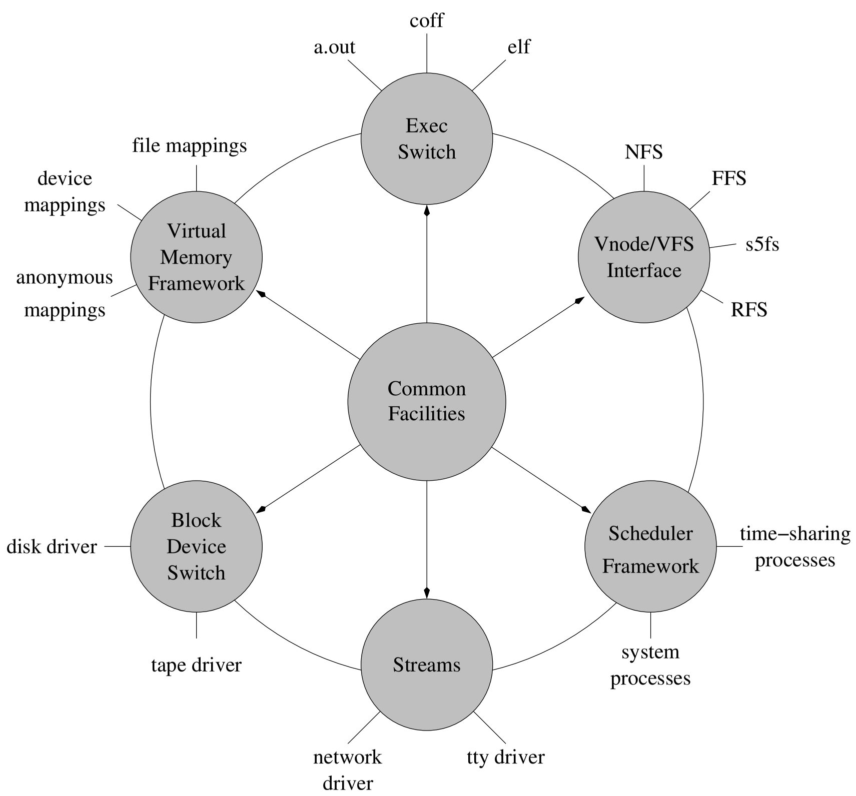


2.7 - Sistema UNIX Moderno

✡ Como resultado da grande diversidade de implementações customizadas, cada qual oferecendo novas funcionalidades, houve a necessidade de uma nova implementação que unifica-se muitas das importantes inovações;

- ... na nova arquitetura, encontramos um pequeno núcleo de facilidades escritas segundo a abordagem modular, que oferece funções e serviços aos processos do sistema operacional; e círculos externos representando funções e interface oferecendo as demais funcionalidades de um sistema operacional;

... 2.7 - Sistema UNIX Moderno



2.7 - Sistema UNIX Moderno

✰ **System V Release 4:** desenvolvido em conjunto pela AT&T e SUN Microsystems, combina aspectos do SVR 3, 4.3 BSD, Microsoft Xenix System V e SunOS;

- ... obtido a partir da reescrita do *kernel* System V com a adição de novas funções: suporte ao processamento de tempo real, classes de escalonamento de processos, estruturas de dados alocadas dinamicamente, gerenciamento de memória virtual, sistema de arquivo virtual e *kernel* preemptivo.

✰ **Solaris 2.x** versão UNIX baseada no SVR4 e produzida pela Sun Microsystems que oferece todas as funcionalidades da versão SVR 4 e outros aspectos avançados tais como: totalmente preemptável, *multithreaded kernel*, suporte para SMP interface do sistema de arquivos orientada a objetos;

- ... é a implementação UNIX mais largamente usada e com o maior sucesso comercial.

2.7 - Sistema UNIX Moderno

✰ **Berkeley Software Distribution 4.4:** produzida em Berkeley, teve um papel crucial no desenvolvimento da teoria de projetos dos sistemas operacionais;

- ... largamente utilizado em instalações acadêmicas, serviu como alicerce para um grande número de produtos UNIX comerciais;

- ... pode-se assegurar que o BSD é o grande responsável pela popularidade do UNIX, bem como pelos aperfeiçoamentos que apareceram primeiramente em suas versões;

- ... os principais aspectos incluíram um novo sistema virtual de memória, mudanças na estrutura do *kernel* e uma longa lista de aspectos inovadores.