

Capítulo 1

Introdução

Neste capítulo vamos apresentar os **sistemas operacionais**, seus objetivos e um breve histórico de sua evolução. Também proporemos uma forma simples para sua classificação e forneceremos alguns outros conceitos importantes.

1.1 Definindo os sistemas operacionais

Desde sua criação, os computadores sempre foram sistemas de elevada sofisticação em relação ao estágio tecnológico de suas épocas de desenvolvimento. Ao longo dos últimos 50 anos evoluíram incrivelmente e, embora tenham se tornado mais comuns e acessíveis, sua popularização ainda esconde sua tremenda complexidade interna.

Neste sentido, os sistemas operacionais, em termos de suas origens e desenvolvimento, acompanharam a própria evolução dos computadores. Deitel nos traz a seguinte definição de **sistema operacional**:

Vemos um sistema operacional como os programas, implementados como software ou firmware, que tornam o hardware utilizável. O hardware oferece capacidade computacional bruta. Os sistemas operacionais disponibilizam convenientemente tais capacidades aos usuários, gerenciando cuidadosamente o hardware para que se obtenha uma performance adequada. [DEI92, p. 3]

Nesta definição surgem alguns novos termos explicados a seguir. O *hardware* é o conjunto de dispositivos elétricos, eletrônicos, ópticos e eletromecânicos que compõe o computador, sendo a máquina física propriamente dita. O *hardware*, aparentemente identificável pelos dispositivos ou módulos que compõe um sistema computacional, determina as capacidades deste sistema. O *software* é o conjunto de todos os programas de computador em operação num dado computador. Já o *firmware* é representado por programas especiais armazenados de forma permanente no *hardware* do computador que

permitem o funcionamento elementar e a realização de operações básicas em certos dispositivos do computador, geralmente associadas a alguns periféricos e a execução de outros programas também especiais.

Na Figura 1.1 podemos identificar o *hardware* como sendo os dispositivos físicos, sua microprogramação e o *firmware* existente neste computador. Como exemplos de dispositivos existentes num sistema podemos citar os circuitos integrados de memória, as unidades de disco flexível ou rígido e o processador do sistema, sendo este último um dispositivo microprogramado. O *firmware* geralmente vem acondicionado em circuitos de memória não volátil (ROM, PROM ou EPROM) sendo os programas ali gravados escritos geralmente em linguagem de máquina e destinados a execução de operações especiais tal como a auto-verificação inicial do sistema (POST ou *power on self test*) e a carga do sistema operacional a partir de algum dispositivo adequado (*bootstrap*).

O *software* deste sistema ou os programas do sistema são representados pelo sistema operacional e todos os seus componentes (bibliotecas de funções e programas utilitários) além de todos os outros programas acessórios do sistema, tais como editores de texto, programas gráficos, compiladores, interpretadores de comando (*shells*), aplicativos de comunicação e ferramentas de administração e manutenção do sistema.

Os programas de aplicação são todos os demais *softwares*, desenvolvidos com finalidades particulares, que são utilizados num dado sistema computacional sob suporte e supervisão do sistema operacional, tais como planilhas eletrônicas, programas de correio eletrônico, navegadores (browsers), jogos, aplicações multimídia etc.

Jogos	Sistemas Específicos	Outros Sistemas
Editores	Compiladores	Shells
Sistema Operacional		
Firmware		
Microprogramação		
Dispositivos Físicos		

Figura 1.1: *Hardware*, *software*, *firmware* e o SO

Por si só, o *hardware* do computador dificilmente poderia ser utilizado diretamente e mesmo assim, exigindo grande conhecimento e esforço para execução de tarefas muito simples. Neste nível, o computador somente é ca-

paz de entender programas diretamente escritos em linguagem de máquina. Além disso, cada diferente tipo de computador possui uma arquitetura interna distinta que pode se utilizar de diferentes processadores que por sua vez requisitarão diferentes linguagens de máquina, tornando penosa e cansativa a tarefa dos programadores.

Desta forma, é adequada a existência de uma camada intermediária entre o *hardware* e os programas de aplicação que pudesse não apenas oferecer um ambiente de programação mais adequado mas também um ambiente de trabalho mais simples, seguro e eficiente.

Um **sistema operacional** é um programa, ou conjunto de programas, especialmente desenvolvido para oferecer, da forma mais simples e transparente possível, os recursos de um sistema computacional aos seus usuários, controlando e organizando o uso destes recursos de maneira que se obtenha um sistema eficiente e seguro.

Stallings, ao tratar dos objetivos e funções dos sistemas operacionais, afirma que:

Um sistema operacional é um programa que controla a execução dos programas de aplicação e atua como uma interface entre o usuário do computador o hardware do computador. Um sistema operacional pode ser pensado como tendo dois objetivos ou desempenhando duas funções: conveniência, pois faz o sistema computacional mais conveniente de usar; e eficiência, pois permite que os recursos do sistema computacional sejam usados de maneira eficiente. [STA96, p. 222]

Silberschatz utiliza praticamente a mesma definição, indicando que um sistema operacional é um ambiente intermediário entre o usuário e o *hardware* do computador no qual programas podem ser executados de forma conveniente e eficiente [SG00, p. 23]. Davis [DAV91], Shay [SHA96] e outros também apresentam idéias semelhantes.

Tanenbaum, por sua vez, define um sistema operacional através de uma ótica ligeiramente diferente:

O mais fundamental de todos os programas do sistema é o sistema operacional que controla todos os recursos computacionais e provê uma base sobre a qual programas de aplicação podem ser escritos. [TAN92, p. 1]

Os sistemas operacionais são uma camada de *software* que "envolve" os componentes físicos de um computador, intermediando as interações entre estes componentes e os usuários ou os programas dos usuários. Neste sentido é apropriado considerar que os sistemas operacionais podem ser vistos como uma extensão do próprio computador ou como gerenciadores dos recursos existentes neste computador.

Ao invés de lidar com a complexidade inerente ao *hardware*, o sistema operacional oferece a funcionalidade disponível no *hardware* através de uma interface de programação orientada a operação de cada tipo de recurso, proporcionando não apenas transparência, mas também isolando o *hardware* das aplicações. Assim temos que o comportamento do sistema operacional é como uma extensão do próprio *hardware* ou como uma máquina virtual, que possui características diferentes da máquina física real.

Imaginando que múltiplos programas em execução desejem fazer uso dos diversos recursos do *hardware*, não é razoável que o controle destes recursos seja transferido aos programadores pois isto acrescentaria uma sobrecarga desnecessária a cada programa, sem que fosse possível otimizar o uso dos recursos. Além disso, erros ou omissões, mesmo que involuntárias, poderiam provocar erros de dimensões catastróficas, acarretando perda de grandes quantidades de dados, violações importantes de segurança etc. O sistema operacional deve se encarregar de controlar os recursos do computador, garantindo seu uso adequado, buscando também otimizar tal uso objetivando um melhor eficiência do sistema, assim sendo, o sistema operacional se comporta como gerente dos recursos do computador.

1.2 Objetivos de um sistema operacional

A despeito do tipo, sofisticação ou capacidades do computador, um sistema operacional deve atender aos seguintes princípios:

1. Oferecer os recursos do sistema de forma simples e transparente;
2. Gerenciar a utilização dos recursos existentes buscando seu uso eficiente em termos do sistema; e
3. Garantir a integridade e a segurança dos dados armazenados e processados no sistema e também de seus recursos físicos.

Além destes objetivos, um sistema operacional também deve proporcionar uma interface adequada para que ele possa ser utilizado pelos seus usuários. Historicamente as primeiras interfaces dos sistemas operacionais eram baseadas em um conjunto de palavras-chave (comandos) e mensagens de diálogo que permitiam a execução de tarefas e a comunicação entre homem (o operador) e máquina. Estes comandos e mensagens definiam a Interface Humano-Computador (IHC) daquele sistema. Atualmente as interfaces baseadas em modo texto estão em desuso, sendo substituídas por interfaces gráficas mais modernas e simples que buscam facilitar a utilização do computador através de sua aparência atraente e uso intuitivo.

1.3 Breve histórico

O desenvolvimento dos sistemas operacionais pode ser melhor observado e compreendido quando acompanhamos a história do próprio computador. No breve resumo que segue pretende-se focar os principais eventos e movimentos relacionados ao desenvolvimento destas máquinas.

1.3.1 O início

Por volta de 1643, Blaise Pascal projeta e constrói uma máquina de calcular mecânica, que deu origem as calculadoras mecânicas utilizadas até meados do século XX. No final do século XVIII, Joseph-Marie Jacquard constrói um tear que utiliza cartões de papelão perfurado para determinar o padrão a ser desenhado no tecido, criando a primeira máquina programável.

Charles Babbage constrói em 1822 a máquina de diferenças e depois, partindo das idéias de Jacquard, projeta a máquina analítica, a qual não foi terminada. Recentemente comprovou-se que suas idéias eram corretas, sendo ele hoje reconhecido como pai do computador moderno. Em 1870 é construído uma máquina analógica para previsão de marés por William Thomson, que origina os computadores analógicos. Em 1890 o Censo Americano utiliza com grande sucesso as máquinas de tabular de Herman Hollerith, que funda em 1896 a *Tabulating Machine Company*. Alguns anos depois esta companhia se transformaria na IBM (*International Business Machines*).

Na década de 30 começam a surgir, em diferentes pontos do mundo, projetos de máquinas eletromecânicas e eletrônicas de calcular:

1934 Máquina eletromecânica programável do engenheiro Konrad Zuse.

1935 Início do projeto da máquina eletrônica ABC, baseada em válvulas, para resolução de sistemas, proposta pelo físico John Vicent Atanasoft.

1937 John Von Neumann, matemático húngaro, propõe uma arquitetura genérica para o computador, utilizada até hoje (veja Figura 1.2).

1939 Desenvolvida a primeira calculadora eletromecânica dos laboratórios Bell.

As descobertas da época motivaram cientistas de diversas especialidades a trabalhar no desenvolvimento dos então chamados cérebros eletrônicos.

1.3.2 Década de 1940

Os primeiros computadores eram realmente grandes máquinas de calcular. Compostas por circuitos baseados em relês e outros dispositivos eletromecânicos, estas máquinas eram muito grandes, lentas, consumiam muita

energia elétrica e eram de difícil operação. Esta tecnologia foi progressivamente substituída pelas válvulas eletrônicas, um pouco mais confiáveis e rápidas, embora muito mais caras. Com isso os computadores da época eram caríssimos, restringindo seu uso à organismos militares, agências governamentais e grandes universidades. O uso do computador era claramente experimental.

Um dos primeiros sistemas programáveis construídos foi o computador eletromecânico Mark I, projetado em conjunto pela IBM e pela Universidade de Harvard, apresentado em 1944. Em 1946 o Exército Americano revela seu computador eletrônico digital, o ENIAC, utilizado para cálculos de balística externa, que vinha sendo utilizado a alguns anos.

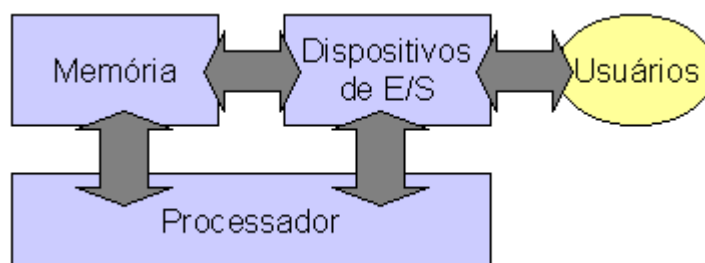


Figura 1.2: Arquitetura de Von Neumann

O uso destes sistemas exigia um grande grau de conhecimento de sua arquitetura e funcionamento. Os engenheiros exerciam o papel de programadores, determinando quais módulos deveriam ser interligados e em que ordem. Um grupo de técnicos treinados, de posse de esquemas de ligação, realizavam a conexão de tais módulos, de forma que a máquina pudesse ser energizada e os cálculos realizados. Nesta época o computador não era programável pois seu *hardware* era modificado para cada tipo de problema diferente, o que representava uma tarefa complexa e delicada.

Nesta década o matemático Von Neumann propôs a construção de sistema computacional baseada numa arquitetura composta por três blocos básicos (Figura 1.2) onde a seqüência de passos a ser executada pela máquina fosse armazenada nela própria sem necessidade de modificação de seu *hardware*. O computador de Von Neumann era uma máquina genérica, cujo bloco processador seria capaz de realizar um conjunto de operações matemáticas e lógicas além de algumas operações de movimentação de dados entre os blocos da máquina. Estas operações, chamadas de instruções, seriam armazenadas no bloco memória enquanto o bloco de dispositivos de E/S (Entrada e Saída) ou I/O (*Input and Output*) seria responsável pela entrada e saída dos dados, instruções e controle do sistema.

A seguir temos a Figura 3 representando o esquema básico de funcionamento dos processadores. Após ligado, um processador efetua um ciclo de busca por uma instrução na memória (*fetch*), a qual é decodificada

do ciclo seguinte (*decode*), que determina quais as ações necessárias para sua execução no último ciclo (*execute*). Após a execução repetem-se os ciclos de busca, decodificação e execução indefinidamente. A repetição desta seqüência só é interrompida através da execução de uma instrução de parada (*halt*) ou pela ocorrência de um erro grave.

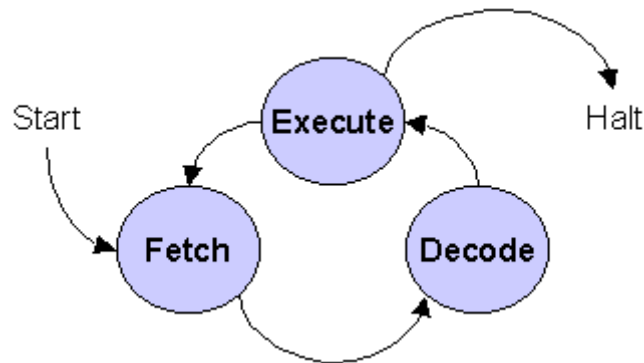


Figura 1.3: Funcionamento básico dos processadores

Assim sendo, um mesmo *hardware* poderia resolver diferentes tipos de problemas sem necessidade de qualquer modificação, bastando que uma seqüência adequada de instruções fosse carregada no computador. Com isto nascem os conceitos de programa, computador programável e com eles a programação de computadores. Apesar de todas as restrições tecnológicas da época, as idéias de Von Neumann revolucionaram a construção dos computadores e ditaram a nova direção a ser seguida.

1.3.3 Década de 1950

A descoberta do transistor deu novo impulso à eletrônica e aos computadores. Apesar de seu custo ainda alto, já era possível fabricar e vender computadores para grandes empresas e organismos governamentais, tanto que em 1951 surge o primeiro computador comercial, o Univac-I (*Universal Automatic Computer*) e em 1953 a IBM lança seu primeiro computador digital, o IBM 701.

Para programá-los ainda era necessário conhecer detalhes sobre seus circuitos e sobre o funcionamento de seus dispositivos. Tais exigências faziam que os computadores só pudessem ser utilizados por especialistas em eletrônica e programação. Mesmo tais especialistas tinham dificuldade em lidar com diferentes computadores, dado que cada computador possuía uma estrutura e funcionamento particulares.

A programação era feita em *assembly*, ou seja, diretamente em linguagem de máquina. Com a evolução dos computadores, tornou-se necessário criar pequenos programas que os controlassem na execução de tarefas cotidianas, tais como acionar certos dispositivos em operações repetitivas ou mesmo

simplificar a execução de novos programas. Surgiram assim os primeiros sistemas operacionais.

O uso individual do computador (conceito de *open shop*) era pouco produtivo, pois a entrada de programas constituía uma etapa muito lenta e demorada que na prática representava o computador parado.

Para otimizar a entrada de programas surgiram as máquinas leitoras de cartão perfurado (semelhantes as máquinas de tabular construídas por Herman Hollerith) que aceleravam muito a entrada de dados. Os programadores deveriam então escrever seus programas e transcrevê-los em cartões perfurados. Cada programa e seus respectivos dados eram organizados em conjuntos denominados *jobs* que poderiam ser processados da seguinte forma: os vários *jobs* de diferentes usuários eram lidos por uma máquina leitora de cartões que gravava os dados numa fita magnética. Esta fita era levada para o computador propriamente dito que lia os *jobs*, um a um, gravando uma outra fita magnética com os resultados de cada job. Esta fita de saída era levada a outro máquina que lia a fita e imprimia as listagens, devolvidas aos usuário juntamente com seus cartões.



Figura 1.4: Sistema Univac, 1951 (processamento em lote - *batch*)

Apesar da natureza seqüencial do processamento, para os usuários era como se um lote de *jobs* fosse processado a cada vez, originando o termo processamento em lote (*batch processing*). Os sistemas *batch* viabilizaram o uso comercial dos computadores, época em que grandes fabricantes de computadores começam a surgir.

Ainda na década de 50 surgiram a primeira linguagem de programação de alto nível (o IBM FORTRAN - *Formula Translator* - em 1957), a primeira unidade de disquetes comercialmente disponível no modelo IBM 305 e os mecanismos de interrupção implementados diretamente no *hardware* dos processadores. Em 1959 a DEC (*Digital Equipment Corporation*) apresenta seu minicomputador, o PDP-I, que fez grande sucesso comercial, originando uma grande linha de equipamentos.

1.3.4 Década de 1960

Buscando uma utilização mais eficiente e segura dos computadores, os sistemas operacionais foram se tornando cada vez mais complexos, passando a administrar os recursos do computador de forma cada vez mais sofisticada. Ao mesmo tempo em que se buscava um uso mais eficiente e seguro do computador, estudavam-se alternativas para que pessoas menos especializadas nos aspectos construtivos da máquina pudessem utilizar o computador, se concentrando em suas verdadeiras tarefas e ampliando as possibilidades de uso dos computadores.

Nesta década aparece o COBOL (*Commom Business Oriented Language*), linguagem de programação especialmente desenvolvida para o Pentágono americano para auxiliar o desenvolvimento de sistemas comerciais. Em 1961 a Farchild inicia a comercialização dos primeiros circuitos integrados. Em 1963 a DEC introduz o uso de terminais de vídeo e no ano seguinte surge o *mouse*. Um dos primeiros avanços ocorridos na década de 60 foi a utilização da multiprogramação. Segundo Deitel:

Multiprogramação é quando vários jobs estão na memória principal simultaneamente, enquanto o processador é chaveado de um job para outro job fazendo-os avançarem enquanto os dispositivos periféricos são mantidos em uso quase constante. [DEI92, p. 4]

Enquanto o processamento chamado científico era muito bem atendido pelo processamento em lote comum o mesmo não acontecia com processamento dito comercial. No processamento científico ocorre a execução de grande quantidade de cálculos com quantidades relativamente pequenas de dados, mantendo o processador ocupado na maior parte do tempo sendo que o tempo gasto com I/O (entrada e saída) era insignificante, daí este comportamento ser chamado *CPU Bounded*. Já no processamento comercial o processador permanece bastante ocioso dado que os cálculos são relativamente simples e o uso de I/O é freqüente dada a quantidade de dados a ser processada, temos um comportamento *I/O Bounded*.

A multiprogramação permitiu uma solução para este problema através da divisão da memória em partes, chamadas partições, onde em cada divisão um *job* poderia ser mantido em execução. Com vários *jobs* na memória o processador permaneceria ocupado o suficiente para compensar o tempo das operações mais lentas de I/O.

A utilização de circuitos integrados na construção de computadores comerciais e a criação de famílias de computadores compatíveis iniciadas com o IBM System/360 inaugurou uma nova era na computação e a expansão de sua utilização. Outra técnica utilizada era o *spooling* (*simultaneous peripheral operation on line*), isto é, a habilidade de certos sistemas operacionais em ler novos *jobs* de cartões ou fitas armazenado-os em uma área temporária

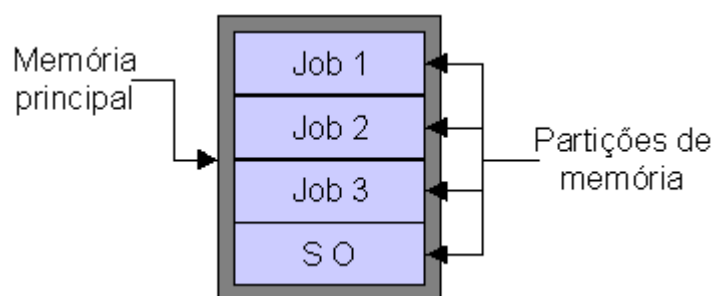


Figura 1.5: Partições de memória num sistema multiprogramado

do disco rígido interno para uso posterior quando uma partição de memória fosse liberada [TAN92, p. 9].

Apesar destas novas técnicas, os sistemas da época operavam basicamente em lote. Assim, enquanto satisfaziam as necessidades mais comuns de processamento comercial e científico, não ofereciam boas condições para o desenvolvimento de novos programas. Num sistema em lote, a correção de um problema simples de sintaxe poderia levar horas devido a rotina imposta: preparação dos cartões, submissão do job no próximo lote e a retirada dos resultados várias horas ou até mesmo dias depois.

Tais problemas associados ao desenvolvimento de *software* motivaram a concepção de sistemas multiprogramados, isto é, sistemas que permitissem o uso simultâneo do computador por diversos usuários através do pseudoparalelismo. O pseudoparalelismo poderia ser obtido com o chaveamento do processador entre vários processos que poderiam atender os usuários desde que fossem dotados de interfaces interativas. A idéia central destes sistemas era dividir o poder computacional de um computador entre seus vários usuários, fornecendo uma máquina virtual para cada um destes. Esta máquina virtual deveria passar a impressão de que o computador estava integralmente disponível para cada usuário, mesmo que isto não fosse verdade.

Nestes sistemas, denominados de sistemas em tempo repartido (*time sharing systems*), o tempo do processador era dividido em pequenos intervalos denominados *quanta* de tempo ou janelas temporais, como ilustra a (Figura 1.6). Tais *quanta* de tempo eram distribuídos seqüencialmente entre os processos de cada usuário, de forma que a espera entre os intervalos fosse imperceptível para os usuários. Depois que um *quanta* de tempo era distribuído para cada processo, se iniciava um novo ciclo de trabalho. A um dado processo seriam concedidos tantos *quanta* de tempo quanto necessário mas apenas um a cada ciclo. Com isto, a capacidade de processamento da máquina ficava dividida entre os usuários do sistema a razão de $\frac{1}{n}$, onde n é o número de usuários do sistema.

O mecanismo central de funcionamento destes sistemas é a interrupção. Uma interrupção é uma forma de parar a execução de um programa qual-

quer, conduzindo o processador a execução de uma outra rotina, previamente especificada que após ser executada por completo, permite o retorno ao programa original, sem prejuízo para execução do primeiro.

Os sistemas em tempo repartido também criaram as necessidades dos mecanismos de identificação de usuário (*userid* e *password*), dos mecanismos de início e término de sessão de trabalho (*login* ou *logon*), da existência de contas de usuário, etc.

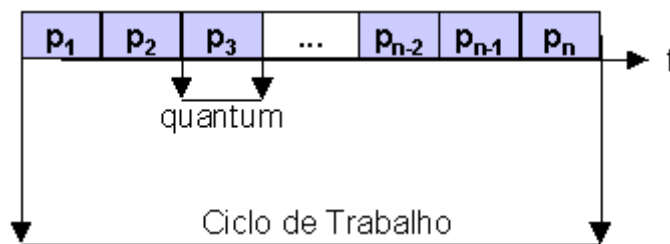


Figura 1.6: Ciclo de trabalho em sistemas de tempo repartido

Um dos processos de cada usuário poderia ser um interpretador de comandos (*shell*) que representaria a interface do computador para o usuário. Inicialmente estas interfaces eram implementadas como linhas de comando em terminais de teletipo e depois em terminais de vídeo. Como utilizavam exclusivamente caracteres também são conhecidas como interfaces em modo texto. As maiores dificuldades que surgiram durante o desenvolvimento destes sistemas foram a solução dos problemas associados ao compartilhamento dos recursos, à organização dos usuários e das tarefas do sistema, além dos mecanismos de segurança e privacidade.

1.3.5 Década de 1970 e 1980

Estas décadas são marcadas especialmente pelo crescimento em tamanho, sofisticação e complexidade dos sistemas computacionais. Aparece o termo *mainframe* (computador principal) e tornam-se cada vez mais comuns os centros de processamento de dados (CPDs) corporativos de uso privativo e bureaus de processamento de dados que vendiam serviços para terceiros. Surge o primeiro microprocessador comercial integrado em um único chip, o Intel 4004 em 1971. Na década de 70 começam a surgir pequenos computadores destinados ao uso pessoal ou doméstico, começando pelos microcomputadores da Xerox em Palo Alto em 1973, o Altair 8.800 destinado ao consumo em massa em 1974, o IBM 5100 em 1975 e o sucesso de vendas do Apple 2 lançado em 1976. Começam a ser desenvolvidas as primeiras aplicações comerciais dos microcomputadores.

A década de 80 viu o surgimento da geração do microcomputadores, o boom dos sistemas desktop a partir do lançamento do IBM-PC (Personal Computer) em 1981 e o desenvolvimento de uma enorme indústria de *hard-*

ware, *software* e serviços originadas nestes eventos. Na mesma década de 80 surgem empresas especializadas em arquiteturas de alto desempenho e são desenvolvidos os primeiros supercomputadores tais como os modelos 1-A ou X-MP da Cray Research.

1.3.6 Década de 1990

Esta década é marcada pelas estações de trabalho (*workstations*), pela computação pessoal portátil e pela interoperabilidade. Dispõe-se de microcomputadores cujo poder de processamento é maior do que os *mainframes* da década de 1970. Os supercomputadores, fabricados quase que artesanalmente, começam a ser tornar máquinas fantásticas, tal como o Cray Y-MP capaz de processar até 2.667 GFlops¹[NCA04]. Tais equipamentos são frequentemente empregados para simulações complexas, tais como as requeridas pela meteorologia, astrofísica, física atômica, farmacêutica, engenharia entre outras.

Os *notebooks*, *palmtops* e PDAs (*personal digital assistants*) representam o máximo em portabilidade e flexibilidade, e estão grandemente incorporados no cotidiano de executivos, profissionais de *design*, vendedores e outros profissionais liberais, mesmo aqueles cuja atividade fim não é a informática. Mesmo estes pequenos aparelhos possuem versões simplificadas de sistemas operacionais, permitindo que uma gama maior de serviços e funcionalidades sejam oferecidas. Além disso a eletrônica embarcada começa a ser tornar presente em automóveis e utensílios domésticos, além das tradicionais aplicações industriais.

Ferramentas de desenvolvimento rápido (RAD tools), ferramentas CASE sofisticadas para modelagem de dados e sistemas, complexos sistemas de CAD-CAE-CAM (Computer Aided Design, Engineering ou Manufacturing). Internet. Sistemas de correio eletrônico, grupos de discussão, educação à distância, multimídia, ATM, Java, Corba, sistemas distribuídos, processamento vetorial, processamento paralelo etc. Inúmeras tecnologias de projeto, desenvolvimento, aplicação, integração e interoperabilidade, adicionando diversidade, novas necessidades e maior complexidade aos sistemas operacionais contemporâneos.

Com isto, alguns milhões de linhas de programa são necessárias para a criação de um sistema operacional que atenda em parte tais necessidades, sendo necessário adaptá-los aos mais diversos tipos de aplicação e usuário. E para dar suporte a tremenda complexidade destes programas, a indústria de *hardware* vem continuamente desenvolvendo novos processadores também com velocidade cada vez maior, circuitos de memória de maior capacidade

¹A capacidade de processamento computacional (*computer power*) é usualmente medida em termos do número de operações em ponto flutuante completas por segundo (Flops).



Figura 1.7: Servidor, *workstation* e *notebook*

e velocidade bem como dispositivos periféricos cada vez mais sofisticados, tudo através das novas tecnologias.

1.3.7 Década de 2000

A supercomputação se torna superlativa. Exemplos disso são os novos sistemas da IBM e da Cray Research, conhecidos como IBM BlueSky e Cray X1 (Figura 1.8) que, respectivamente, tem capacidades de processamento de 6.323 TFlops [NCA04] e 52.4 TFlops [CRA04]. Graças à pesquisa e a evolução dos sistemas distribuídos, tal como as técnicas de *grid computing*, o emprego de *clusters* (agrupamentos) de computadores geograficamente dispersos e interligados através da Internet ou conectados através de redes locais de alto desempenho também vem se tornando mais comum, como alternativa mais econômica e de desempenho semelhante aos supercomputadores.

O desenvolvimento e miniaturização de novos componentes eletrônicos em adição e a utilização de novos processos de fabricação associados à grande expansão das redes digitais de telefonia móvel permitem que muitos milhões de usuários se beneficiem de equipamentos que somam as características de telefones portáteis com PDAs cada vez mais sofisticados.

Os computadores dedicados e os de uso genérico estão se tornando cada vez mais populares em contraste com as necessidades e exigências mais sofisticadas de seus usuários. O número de usuários de computador também cresce vertiginosamente, impulsionando os fabricantes a criarem novos produtos, serviços e soluções.

Com tudo isso a computação é, sem dúvida alguma, imprescindível como ferramenta para o homem moderno, tornando o desenvolvimento contínuo dos sistemas operacionais uma tarefa essencial.

1.4 Tipos de sistemas operacionais

Identificamos através da história dos sistemas operacionais alguns tipos de sistemas operacionais, os quais são comparados segundo alguns aspectos considerados importantes como pode ser visto na Tabela 1.1.



Figura 1.8: Supercomputador Cray X1

Tabela 1.1: Tipos de sistemas operacionais

Tipo de SO	Interativo	Tempo de Resposta	Produtividade (Throughput)	Multiusuário
Open Shop	Sim	Baixo Irregular	Baixa	Não
Batch Simples	Não	Alto Regular	Média Alta	Sim
Batch com Spooling	Não	Alto Regular	Média Alta	Sim
Tempo Repartido	Sim	Baixo Previsível	Média	Sim
Tempo Real	Sim	Baixo Previsível	Média	Sim

A **interatividade** é o aspecto que considera se o usuário utiliza diretamente o sistema computacional, podendo receber as respostas deste, sem intermediação e dentro de intervalos de tempo razoáveis.

O **tempo de resposta** (*response time*) é, desta forma, uma medida de interatividade, que representa o intervalo de tempo decorrido entre um pedido ou solitação de processamento (por exemplos, a entrada de um comando ou execução de um programa) e a resposta produzida pelo sistema (realização das operações solicitadas ou finalização do programa após sua execução completa). Tempos de resposta da ordem de alguns segundos configuram sistemas interativos, embora sejam admitidas esperas mais longas. Por exemplo, uma resposta produzida num intervalo de 30 segundos pode ser considerada normal levando-se em conta a natureza e complexidade da operação solicitada. Embora normal, intervalos desta ordem de grandeza ou superiores tornam enfadonho e cansativo o trabalho com computadores.

O **tempo de reação** (*reaction time*) também é outra medida de interatividade a qual considera o tempo decorrido entre a solicitação de uma ação e seu efetivo processamento.

Já a **produtividade** (*throughput*) é uma medida de trabalho relativa do sistema, expressa usualmente em tarefas completas por unidade de tempo, ou seja, é uma medida que relaciona o trabalho efetivamente produzido e o tempo utilizado para realização deste trabalho. Unidades possíveis do *throughput* são: programas por hora, tarefas por hora, *jobs* por dia etc.

A produtividade não deve ser confundida com o desempenho bruto do processador do sistema (sua capacidade de processamento), pois depende muito da arquitetura do sistema e do sistema operacional o quanto desta capacidade é efetivamente convertida em trabalho útil e o quanto é dispendida nas tarefas de controle e gerência do próprio sistema computacional.

Podemos notar que dentre estas medidas de performance (existem ainda diversas outras), algumas são orientadas ao usuário, tal como o tempo de resposta ou o tempo de reação; enquanto outras são orientadas ao sistema em si, tal como a taxa de utilização do processador ou a produtividade [DEI92, p. 423].

1.5 Recursos e ambiente operacional

O *hardware* do computador, ou seja, sua parte física, determina suas capacidades brutas, isto é, seus verdadeiros limites. Todos os elementos funcionais do computador são considerados **recursos** do sistema computacional e são, geralmente, representados pelos dispositivos que o compõe e que podem ser utilizados pelos usuários, ou seja: monitores de vídeo, teclado, *mouse*, mesas digitalizadoras, portas de comunicação serial e paralela, placas de rede ou comunicação, impressoras, *scanners*, unidades de disco flexível ou rígido, unidades de fita, unidades leitoras/gravadoras de CD, DVDs etc.

O sistema operacional aparece como uma camada sobre o *hardware* e *firmware*, mas simultaneamente envoltória deste. O sistema operacional está sobre o *hardware* e *firmware* pois deles depende para sua própria execução. Ao mesmo tempo é uma camada envoltória pois pretende oferecer os recursos do computador ao usuário do sistema minimizando os aspectos de como são tais dispositivos ou como serão feitas as operações que os utilizam. Desta forma o sistema operacional, através de sua interface, define uma nova máquina que é a combinação de um certo *hardware* com este sistema operacional.

O conjunto de *hardware* e sistema operacional, usualmente chamado de **plataforma** ou **ambiente operacional**, é aparentemente capaz de realizar tarefas de um modo específico ditado pela própria interface. Note que o ambiente operacional é distinto do *hardware*, pois o *hardware* do computador, por si só, não é capaz de copiar um determinado arquivo de uma unidade

de disco rígido para uma unidade de disquete. Para realizar esta cópia, uma série de procedimentos devem ser executados, indo desde o acionamento das unidades utilizadas, passando pela localização das partes do arquivo origem e das áreas disponíveis no disquete de destino, até a transferência efetiva dos dados.

Através da interface do sistema operacional, tal ação poderia ser possível através de um comando fornecido dentro de um console (uma interface de modo texto, tal como mostra a Figura 1.9, por exemplo:

```
copy so.dvi a:\
```

ou, se dispusermos de uma interface gráfica (Figura 1.10), através de uma operação visual frequentemente denominada *arrastar e soltar* (*drag and drop*) devido as ações realizadas com o *mouse*.

```
Prompt do MS-DOS: Miktex
<imagens/Note2Server.png> [15] <imagens/Cray81.png> [16]
Underfull \vbox (badness 10000) has occurred while \output is active [17]
Overfull \hbox (6.64293pt too wide) in paragraph at lines 524--527
[]\OTI/cw/r/n/10.95 opera[ç]ao vi-sual fre-quentemente de-not-i-nada arrastar-e
-soltar <drag-and-drop>
<imagens/space300x200.png> <imagens/space300x200.png> [18]
Overfull \hbox (2.41502pt too wide) in paragraph at lines 357--358
[]\OTI/cw/r/n/10.95 Determina[ç]ao de um am-bi-ente de tra-balho equiv-a-lente
para os usuarios,
[19] <so_processos.tex [20]
Cap[itu]lo 2.
> <so_escalonamento.tex [21] [22]
Cap[itu]lo 3.
> <so_memoria.tex [23] [24]
Cap[itu]lo 4.
> <so_io.tex [25] [26]
Cap[itu]lo 5.
> <so_rl.tex [27] [28]
Ap[en]dix A.
> [29] [30] <so.bb1> [31] <so.aux> )
<see the transcript file for additional information>
Output written on so.dvi (41 pages, 77504 bytes).
Transcript written on so.log.
C:\1\Escolas\Faj\SO\texto>copy so.dvi a:\
```

Figura 1.9: Interface em modo texto (linha de comando ou console)

Enquanto o *hardware* isolado não permitia a realização direta da tarefa de copiar um arquivo, através do sistema operacional temos a impressão de que o computador se tornou capaz de realizar a tarefa desejada através do comando dado ou da operação visual efetuada, sem a necessidade do conhecimento de detalhes de como a tarefa é verdadeiramente realizada. A aparentemente expansão das capacidades do *hardware* do computador é, na verdade, uma simplificação obtida através da automatização da tarefa através de sua programação no sistema operacional. O *hardware* é realmente o realizador da tarefa, enquanto o sistema operacional apenas intermedia esta operação através de sua interface.

Observamos que a criação de uma interface, gráfica ou não, para um sistema operacional é extremamente importante ², pois determinam a criação de um **ambiente operacional** consistente. Tais possibilidades são bastante atraentes, pois se cada sistema operacional tem a capacidade de "transformar" o *hardware* de um computador em um determinado ambiente operacional, isto equivale a dizer que computadores diferentes, dotados de um mesmo sistema operacional, oferecerão transparentemente um ambiente operacional

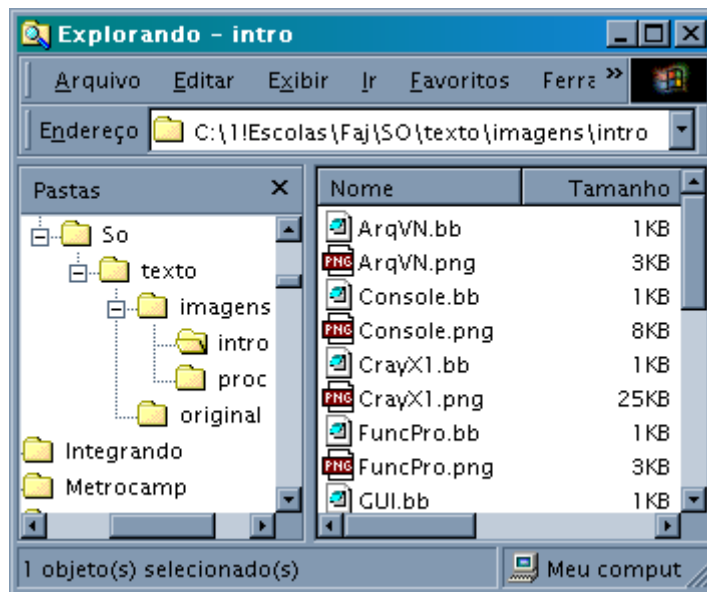


Figura 1.10: Interface gráfica

idêntico, ou seja, se comportarão como se fossem máquinas de um mesmo tipo. Por exemplo, o emprego de uma distribuição do Linux em microcomputadores baseados em processadores na Intel oferece o mesmo ambiente que *workstations* IBM utilizando este mesmo sistema operacional.

Esta possibilidade propicia as seguintes vantagens:

1. Determinação de um ambiente de trabalho equivalente para os usuários, que podem desconhecer as diferenças entre os diversos computadores dotados de um mesmo sistema operacional (plataforma de operação).
2. Criação de um ambiente de desenvolvimento semelhante, onde outros programas podem ser desenvolvidos, testados e utilizados em diferentes computadores dotados de um mesmo SO, com a conseqüente criação de uma plataforma de desenvolvimento.
3. Redução das necessidades de treinamento e aceleração do processo de familiarização e aprendizado no novo ambiente.

Para estudarmos os sistemas operacionais devemos primeiro nos familiarizar com o conceito de processo, fundamental para uma compreensão mais ampla dos problemas envolvidos e de suas soluções. Depois serão estudadas a administração da execução dos programas no sistema, o gerenciamento de memória e também dos dispositivos de entrada e saída.

²O projeto de interfaces consistentes, convenientes e uso simplificado, bem como sua avaliação, é uma disciplina compreendida como IHC ou Interface Humano-Computador.