

Sistemas Operacionais

I

Índice

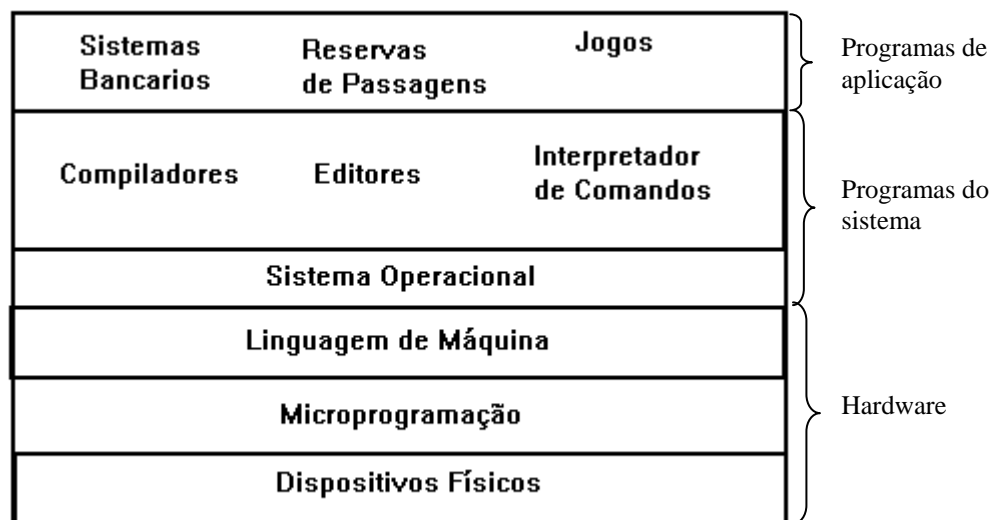
1	INTRODUÇÃO	1
1.1	Definição.....	1
1.1.1	S.O. como Máquina Virtual ou Máquina Estendida	2
1.2	Histórico.....	3
1.2.1	Válvulas e Painéis de Conexão (1945-1955)	3
1.2.2	Transistores e Sistemas de Lote (batch) (1955-1965).....	4
1.2.3	Circuitos Integrados e Multiprogramação (1965-1980).....	5
1.2.4	Computadores Pessoais e Redes (1980 - presente).....	6
1.3	Tipos de Sistemas Operacionais	7
1.3.1	Sistemas operacionais de grande porte	7
1.3.2	Sistemas operacionais de servidores	8
1.3.3	Sistemas operacionais de multiprocessadores.....	8
1.3.4	Sistemas operacionais de computadores pessoais.....	8
1.3.5	Sistemas operacionais de tempo real	8
1.3.6	Sistemas operacionais embarcados	8
1.3.7	Sistemas operacionais de cartões inteligentes.....	9
1.4	Conceitos básicos de Hardware	9
1.4.1	UCP (Unidade Central de Processamento)	9
1.4.2	Clock.....	9
1.4.3	Registradores.....	9
1.4.4	Memória Principal (MP)	10
1.4.5	Memória Cache	10
1.4.6	Memória Secundária (MS).....	10
1.4.7	Dispositivo de Entrada/Saída	10
1.4.8	Barramento.....	10
1.5	Conceitos Básicos de Software	11
1.5.1	Linguagem de Máquina (formato binário).....	11
1.5.2	Tradutor.....	11
1.5.3	Interpretador.....	12
1.5.4	Linker	12
1.5.5	Loader	12
1.5.6	Depurador (debugger)	12
1.5.7	Linguagem de controle.....	13

1 INTRODUÇÃO

1.1 Definição

Qualquer pessoa que teve algum contato com um sistema de computação sabe que o mesmo consiste em dois componentes principais: o hardware (que engloba toda parte fisicamente montada, em geral em circuitos eletrônicos) e o software (que compreende toda a programação para levar o hardware a executar o que foi determinado pelo usuário). É óbvio que a falta de qualquer um dos componentes acima acarreta a inutilidade do outro, bem como a ineficiência de um tem implicações sérias na utilização do outro.

O que talvez não seja de conhecimento difundido é o fato de que, nos sistemas atuais, tanto o hardware como o software apresentam uma organização aproximada em camadas, de forma a facilitar a sua utilização. É possível apresentar um esquema para essa organização como na figura abaixo:



Na parte de hardware, a linguagem de máquina é o que o usuário “enxerga” do processador, isto é, o seu único modo de operar com o processador é através da linguagem de máquina. Ao chegar no processador uma instrução de máquina, a microprogramação é a responsável pela interpretação e controle da sua execução, através da ativação dos dispositivos físicos. Em alguns processadores pequenos a interpretação e controle de execução são também realizados por meio de dispositivos, deixando, portanto de existir a camada de microprogramação. Na parte de Software, o objetivo final são os programas aplicativos. Para possibilitar o desenvolvimento de programas aplicativos existem programas como compiladores, editores de texto, depuradores, entre outros.

Sistemas Operacionais

Estes programas não deixam de ser aplicativos, apenas são voltados ao programador e não ao usuário final. Para permitir a utilização eficiente dos recursos de hardware, bem como fornecer facilidades extras ao usuário existe o sistema operacional.

Assim podemos definir Sistema Operacional como sendo um conjunto de rotinas executado pelo processador, assim como os programas dos usuários. Sua principal função é controlar o funcionamento do computador, gerenciando os vários recursos do sistema. Além disso, fornece a base sobre a qual os programas aplicativos são escritos.

Pode-se dividir as funções do S.O. em duas categorias: a definição de uma máquina virtual e o gerenciamento de recursos.

1.1.1 S.O. como Máquina Virtual ou Máquina Estendida

Como formador de uma máquina virtual, a função do S.O. é apresentar ao usuário uma máquina com as seguintes características:

- i) Facilidade de operação: o S.O. deve fornecer uma interface entre o usuário e o hardware que apresente maior facilidade de programação. Um exemplo típico disto é a escrita em disco flexível. Para esta operação, um controlador precisa das seguintes instruções:
 - a. Recalibração: corresponde a um ajuste da cabeça de leitura na primeira trilha, pois cada movimento posterior da cabeça é sempre relativo, de forma que se a mesma se apresenta inicialmente mal posicionada, todos os futuros posicionamentos serão errados;
 - b. Movimento da cabeça: o deslocamento da mesma para a trilha requerida;
 - c. Espera do setor: que representa uma espera até que a cabeça de leitura se posiciona sobre o setor;
 - d. Escrita dos dados - verificação: para garantir que os dados foram verdadeiramente escritos e sem nenhum erro.

Além destes passos podem existir outros, dependendo do controlador utilizado, do tipo de acesso (leitura, escrita, formatação, etc) e de se ocorrem erros ou não. Atualmente existem diversos controladores de disco que cuidam eles mesmos de tarefas como posicionamento da cabeça, leitura ou escrita, verificação de erros, etc. No entanto mesmo nestes casos o processo de controle restante ao processador central é complicado, pois exige tarefas como:

- a. Controle das trilhas e setores físicos onde se encontra ou deve ser colocada a informação;

Sistemas Operacionais

- b. Tentar novamente no caso de ocorrência de erros, visto que os mesmos são bastante frequentes nos meios magnéticos atuais devido a condições transitórias.

Como vemos, o programador médio não deve ser envolvido com os detalhes deste processo todo. Para isto, os sistemas operacionais fornecem métodos muito mais simples e estruturados de uma forma mais voltada aos problemas do usuário do que é estrutura interna do computador.

- ii) Extensão das capacidades da máquina: o S.O. pode fornecer também algumas capacidades não presentes no computador original, como, por exemplo, múltiplos usuários e sistemas de proteção de acesso.

1.1.2 S.O. como Gerenciador de Recursos

Como um gerenciador de recursos, a função do S.O. é controlar (ou gerenciar) a utilização de todos os recursos fornecidos pelo hardware e a sua distribuição entre os diversos programas que competem por ele de forma a garantir:

- i) A execução correta dos diversos programas;
- ii) Alta eficiência na utilização dos recursos.

Dentro deste ponto de vista então, ao S.O. compete cuidar de quem está usando que recurso, aceitar (ordenadamente) requisições de um dado recurso, realizar a contagem de utilização de recursos e mediar conflitos nos pedidos de recursos por vários programas.

1.2 Histórico

Para uma melhor idéia do curso de desenvolvimento dos sistemas operacionais atuais, apresentamos a esquematização da evolução histórica dos mesmos, enfatizando a relação entre a evolução dos S.O. e os avanços em hardware.

1.2.1 Válvulas e Painéis de Conexão (1945-1955)

Os primeiros computadores foram implementados através de válvulas a vácuo, consistindo em salas inteiras de circuito, consumindo energia elétrica suficiente para cidades inteiras. A programação era realizada através de painéis onde as conexões realizadas representavam os 0 e 1 dos códigos binários da linguagem de máquina. Não existia o conceito de sistema operacional, sendo que cada usuário introduzia o seu programa por painéis e aguardava os resultados. A

Sistemas Operacionais

probabilidade de falha do sistema durante a execução de algum programa era altíssima, devido à baixa confiabilidade das válvulas a vácuo.

1.2.2 Transistores e Sistemas de Lote (batch) (1955-1965)

A introdução dos transistores, com a conseqüente redução de tamanho e consumo e aumento da confiabilidade permitiu o desenvolvimento dos primeiros sistemas realmente utilizáveis fora dos círculos acadêmicos e governamentais, o que garantiu a comercialização dos mesmos.

Nesta época, começa a surgir a distinção entre projetistas, construtores, operadores, programadores e pessoal da manutenção.

Entretanto, os computadores ainda eram extremamente grandes e caros, devendo ser acondicionados em grandes salas com ar condicionado e operados por pessoal profissional. Seu uso era, portanto, restrito a entidades governamentais, grandes corporações e universidades.

O processo de execução de uma tarefa (*job*) era, resumidamente:

- i) Perfuração de um conjunto de cartões com o programa a ser executado;
- ii) O operador pega os cartões e os coloca na leitora. Se o compilador FORTRAN for necessário, ele é colocado (também como um conjunto de cartões) na leitora;
- iii) O resultado sai na impressora e é levado pelo operador para um local onde o usuário o recolhe.

Este processo, além de lento, desperdiça muito tempo de programação devido aos deslocamentos do operador pela sala, buscando conjuntos de cartões a serem utilizados e pela lentidão dos dispositivos de entrada e saída (leitora de cartões e impressora). Para maximizar a eficiência na utilização dos processadores, e devido ao surgimento das unidades de fita magnética, foi utilizada um novo procedimento:

- i) Perfuração dos cartões e envio ao operador;
- ii) O operador junta os conjuntos de cartões e, com a utilização de um computador mais barato, grava-os em uma fita magnética;
- iii) A fita magnética é levada ao processador principal e lida;
- iv) Os programas da fita são executados e o resultado é gravado em outra fita magnética;
- v) Esta fita de saída é levada ao computador secundário (mais barato), lida e seu conteúdo impresso em uma impressora comum;
- vi) A saída da impressora é entregue aos usuários.

Este processo, denominado OFF-LINE, garantiu uma maior eficiência na utilização do processador principal. Porém aumentou o tempo de resposta do sistema para cada usuário. Este aumento do tempo de resposta do sistema se dá em função de se juntar uma quantidade razoável de

Sistemas Operacionais

conjuntos de cartões para se gravar uma fita. Desta forma, cada usuário, para obter a resposta a seu programa, deve aguardar a execução de diversos outros programas armazenados na mesma fita. Isto fica ainda mais crítico quando um dos programas de uma fita apresenta um tempo de execução muito elevado.

1.2.3 Circuitos Integrados e Multiprogramação (1965-1980)

Com a introdução de circuitos integrados, houve uma grande redução no tamanho e custo dos sistemas, bem com um aumento em sua complexidade e generalidade. Isto permitiu o desenvolvimento de dispositivos de entrada e saída inteligentes, de forma que os próprios se responsabilizam pelo controle da transferência de dados entre eles e a memória principal. Outro desenvolvimento importante foi a introdução dos discos, que permitem um acesso aleatório à informação contida nos mesmos, diferentemente das fitas magnéticas, que somente permitem um acesso aos dados na ordem em que os mesmos estão gravados (note que isto pode ficar transparente através de uma programação cuidadosa, entretanto com alto custo em tempo de execução). Estes foram fatores fundamentais para o sucesso do conceito de multiprogramação, apresentado a seguir.

Simultaneamente com a utilização de circuitos integrados, surgiu o conceito de multiprogramação. A idéia provém dos seguintes fatos:

- i) Durante a execução de programas que realizam alta utilização de cálculos (ex: programas científicos) todos os dispositivos de entrada e saída permanecem inativos;
- ii) Durante a execução de programas com alta utilização de entrada e saída (ex: programas comerciais com consultas a base de dados e impressão de relatórios) o processador permanece durante grande percentagem do tempo aguardando os dispositivos de entrada/saída.

Desta forma, surgiu a idéia de se colocar diversas tarefas (*jobs*) dentro de alguma partição da memória principal (*partição* é um trecho de memória alocado a um *job*) e executá-las simultaneamente de forma que, se alguma tarefa precisa aguardar a transferência de dados para um dispositivo, outra tarefa pode utilizar o processador central neste período.

Outro conceito introduzido foi o de *SPOOL* (de “*Simultaneous Peripheral Operation On Line*”) que corresponde à leitura imediata dos *jobs* para o disco no momento da sua chegada, sendo que ao terminar um dos *jobs* ativos, um novo *job* é imediatamente carregado do disco para a partição de memória vazia e executado. Este processo tem a vantagem de que, com a leitura simultânea dos dados para um meio de armazenamento mais rápido e com a transferência de dados entre os meios, realizada simultaneamente com a operação da unidade de processamento principal, desapareceu praticamente o tempo manual de montagem e desmontagem de fitas. Além disso,

Sistemas Operacionais

dispondo de diversos *jobs* a serem executados no disco, o sistema operacional podia escolher entre eles por prioridade, e não necessariamente por ordem de chegada.

Entretanto, até este ponto, o sistema continuava sendo um sistema de lotes, sendo o tempo entre a apresentação de um conjunto de cartões e a retirada do resultado extremamente alto, principalmente quando se está realizando a depuração de programas. Para diminuir o tempo de resposta do sistema a um dado *job* foi introduzido o conceito de compartilhamento de tempo (“*time-sharing*”), no qual cada usuário possui um terminal ligado em linha com o computador, podendo ainda o computador executar, no fundo, alguns lotes com a utilização do tempo disponível devido à lentidão de entrada de dados dos usuários.

Nesta época também surgiram os minicomputadores, com uma menor capacidade de processamento numérico, mas também um custo muito menor.

Obs: **Multiprogramação** e **Multiprocessamento**: Estes conceitos devem ser claramente distinguidos.

- ✓ Multiprogramação: corresponde a diversos programas distintos executando em um mesmo processador.
- ✓ Multiprocessamento: corresponde a diversos processadores, dentro de um mesmo sistema de computação, executando programas diversos ou cooperando na execução de um mesmo programa.

Note que foi a existência de multiprocessamento entre os dispositivos de entrada/saída e o processador central que tornou atrativa a introdução da multiprogramação, mas a relação pára por aí.

1.2.4 Computadores Pessoais e Redes (1980 - presente)

Com a integração em larga escala e o surgimento dos microcomputadores, surge também o conceito de “*user-friendly*” para S.O., que corresponde ao desenvolvimento de sistemas operacionais para serem utilizados por pessoas sem nenhum conhecimento de computação e que, provavelmente, não têm nenhum interesse em vir a conhecer algo.

Um outro desenvolvimento interessante que foi bastante impulsionado pelos microcomputadores (apesar de não depender dos mesmos) é o de sistemas operacionais para redes de computadores, que consistem em computadores distintos interligados por elementos de comunicação. Os sistemas operacionais para redes são divididos em duas categorias:

- i) Sistemas operacionais de rede: no qual cada usuário tem conhecimento de seu próprio computador e pode acessar dados em outros computadores;

Sistemas Operacionais

- ii) Sistemas operacionais distribuídos: em que o sistema operacional faz com que todos os computadores da rede formem uma unidade, de forma que nenhum usuário tenha conhecimento de quantos computadores há na rede ou em qual (ou quais) computador o seu específico programa está executando.

1.3 Tipos de Sistemas Operacionais

Com essa evolução surgiram muitos tipos de sistemas operacionais, alguns conhecidos outros nem tanto. Nesta seção discutiremos resumidamente sobre alguns deles.

1.3.1 Sistemas operacionais de grande porte

Os sistemas operacionais de grande porte diferenciam-se dos computadores pessoais pela capacidade de dispositivos de E/S. Por exemplo, podemos citar a enorme diferença entre a quantidade de discos e os milhares de *gigabytes* de dados que um computador de grande porte pode armazenar, enquanto um computador pessoal armazena algumas centenas de *gigabytes* apenas.

Os sistemas operacionais para os computadores de grande porte oferecem os serviços em lote (*batch*), processamento de transações e tempo compartilhado.

- i) Sistema em lote (*batch*): Processa *jobs* de rotina sem a interação do usuário. Os programas são armazenados em discos ou fitas, onde esperam para serem executados seqüencialmente.

Vantagem: boa utilização do processador.

Desvantagem: tempos de resposta muito longos, em face do processamento puramente seqüencial.

- ii) Processamento de transações: Administram grandes quantidades de pequenas requisições.

- iii) Sistema de tempo compartilhado: permitem que vários usuários remotos executem seus *jobs* simultaneamente no computador.

Vantagem: permite a interação dos usuários com o sistema, geralmente de forma rápida; processador, memória e periféricos são compartilhados;

Desvantagem: implementação complexa.

Sistemas Operacionais

1.3.2 Sistemas operacionais de servidores

Sistemas executados em servidores (computadores pessoais muito grandes, estações de trabalho ou computadores de grande porte). Servem múltiplos usuários de uma vez em uma rede e permitem compartilhar recursos de hardware e software. Ex: Linux.

1.3.3 Sistemas operacionais de multiprocessadores

Sistemas para gerenciamento de várias CPUs conectadas em um único sistema, compartilhando recursos e aumentando o poder de execução dos processos. Esses sistemas também são denominados computadores paralelos, multicomputadores ou multiprocessadores. Ex: Unix.

1.3.4 Sistemas operacionais de computadores pessoais

O único objetivo desse sistema é fornecer uma boa interface para um único usuário. Esses sistemas são usados para digitação de textos, acesso à Internet, planilhas de cálculos e execução de outros aplicativos. Ex: Windows.

1.3.5 Sistemas operacionais de tempo real

Tem como principal fator o tempo como parâmetro fundamental. Existem dois tipos de sistemas de tempo real: o sistema de tempo real crítico e o sistema de tempo real não crítico.

O sistema de tempo real crítico é aquele em que as ações precisam ocorrer nos momentos previamente determinados. Ex: Linha de montagem.

Já no sistema de tempo real não crítico o descumprimento ocasional de uma ação não ocasiona uma falha tão grave ao bom funcionamento das tarefas. Ex: Sistema de áudio.

1.3.6 Sistemas operacionais embarcados

Sistemas de computadores que cabem em pequenos aparelhos, como exemplos temos: PDA, aparelhos de TV, microondas, etc. Principais características são as restrições que possuem referente ao tamanho, memória e consumo de energia. Ex: PalmOS.

Sistemas Operacionais

1.3.7 Sistemas operacionais de cartões inteligentes

Cartões inteligentes são dispositivos do tamanho de cartões de crédito que contém um chip de CPU. Também possuem restrições de consumo de energia e de memória. Podem realizar pagamentos eletrônicos, bem como outras tarefas mais aprimoradas.

Geralmente o sistema operacional residente dentre desses cartões é o JavaOS, juntamente com uma máquina virtual Java (JVM). Ambos ficam armazenados na ROM do cartão inteligente.

1.4 Conceitos básicos de Hardware

1.4.1 UCP (Unidade Central de Processamento)

A UCP (CPU em português) é utilizada para executar todos os programas do sistema. Esse componente é usado pela busca das instruções na memória principal (MP).

A Unidade de Controle (UC) é um dos componentes da UCP. Ela é responsável por controlar as atividades de todos os componentes do sistema.

Outro componente pertencente à UCP é a ULA (Unidade Lógica e Aritmética), responsável pelas operações lógicas (testes e comparações) e aritméticas.

1.4.2 Clock

O Clock está localizado na UCP e é utilizado para gerar pulsos elétricos constantes (periódicos), os quais são utilizados pela UC para execução das instruções.

1.4.3 Registradores

Os registradores são utilizados para armazenamento temporário de informações. Todo computador tem alguns registradores que são especiais pela sua funcionalidade, alguns deles são:

- PC (Program Counter): Contador de instrução utilizado para armazenar o endereço da próxima instrução que a UCP deverá executar;
- SP (Stack Pointer): O apontador de pilha serve para guardar o endereço de memória do topo da pilha (informação das tarefas que tiveram que ser interrompidas).

Sistemas Operacionais

1.4.4 Memória Principal (MP)

A memória principal é utilizada para armazenamento temporário das instruções e dos dados. Utiliza como unidade básica o Bit, o qual pode ser 0 ou 1. O acesso às informações são realizadas através do endereço. Exemplo: Memória RAM.

1.4.5 Memória Cache

A memória cache difere da memória principal pelo tamanho e pelo tempo de acesso, sendo mais rápida que a MP. Quando uma determinada informação deverá ser usada pelo processado, essa informação é transferida da MP para a memória cache.

1.4.6 Memória Secundária (MS)

A memória secundária é um meio de armazenamento de dados e informações permanente, ou seja, quando o computador for desligado as informações armazenadas nesta memória não serão perdidas.

A grande desvantagem na utilização dessa memória é a velocidade de acesso que é bem mais lento do que a MP. Como vantagem temos o baixo custo e a maior capacidade de armazenamento das informações. Exemplos dessas memórias são: a fita magnética, o disco magnético, disco óptico e outros.

1.4.7 Dispositivo de Entrada/Saída

Os dispositivos de entrada e saída fazem a comunicação da UCP e MP com usuários e MS. São dispositivos utilizados como memória secundária (discos e fitas magnéticas), dispositivos de interface (comunicação) homem-máquina, tais como teclado, monitores de vídeo, impressoras, mouses, scanners, microfone, câmeras, etc.

1.4.8 Barramento

Os barramentos são linhas de comunicação entre UCP, MP e dispositivos de E/S. Cada linha é um fio paralelo que trafega informações, como dados, endereços e sinais de controle.

1.5 Conceitos Básicos de Software

Softwares podem ser divididos em: serviços do SO (compiladores, interpretadores, linkers, etc); softwares desenvolvidos pelos usuários (editores de texto, planilhas, sistemas de banco de dados, etc).

1.5.1 Linguagem de Máquina (formato binário)

A linguagem de programação que o processador realmente consegue entender, não requerendo qualquer tipo de tradução ou relocação.

Os programas escritos em linguagem de máquina são programas longos com mais chances de erros e de difícil entendimento para o usuário.

Cada processador possui um único conjunto de instruções de máquina (registradores, modos de endereçamento, tipos de dados), caracterizando a si mesmo e suas potencialidades. Como cada computador tem um único conjunto de instrução, os programas gerados por essas linguagens não são portáteis para diferentes tipos de computadores.

1.5.2 Tradutor

O tradutor traduz os programas-fonte (em linguagem de máquina ou linguagem de alto nível), que ainda não estão prontos para serem executados pela UCP, para código de máquina.

Dentro do tradutor temos o montador e o compilador.

O montador utiliza uma linguagem de montagem, a qual é um conjunto de símbolos associados às instruções da linguagem de máquina do processador. Ele está diretamente ligado às características da arquitetura do processador (portanto, diferente para cada computador).

O compilador é utilizado para verificar erros em um programa escrito em uma linguagem de alto nível, como Pascal, C, Cobol, etc. O compilador não precisa ter nenhuma preocupação com a máquina, sendo então mais portátil, pois é independente de hardware.

Sistemas Operacionais

1.5.3 Interpretador

A partir de um programa em linguagem de alto nível, traduz cada instrução e a executa logo em seguida, no momento da execução do programa.

Quando se utiliza um interpretador para executar um programa, não há a necessidade de se gerar um programa executável, então, podemos concluir que não existe um programa executável para um interpretador.

A grande desvantagem em se utilizar um interpretador é o tempo gasto na tradução das instruções que é muito alto.

A implementação de tipos de dados dinâmicos (mudança do tipo durante a execução do programa) é considerada uma vantagem.

Exemplos de interpretadores são Basic, dBase, APL, etc.

1.5.4 Linker

O linker é utilizado para gerar um programa executável a partir de um ou mais módulos-objeto. As principais funções são: referência a símbolos (consulta em bibliotecas) e relocação de memória (determinar uma região da memória na qual o programa será executado).

1.5.5 Loader

O loader é responsável por carregar os programas na memória. Os dois tipos de loaders são: loader absoluto, o qual determina o endereço inicial da memória e o tamanho; e loader realocável, carrega o programa em qualquer lugar da memória.

1.5.6 Depurador (debugger)

O depurador permite ao usuário controlar toda a execução do programa a fim de detectar erros na estrutura.

Este permite acompanhar a execução do programa instrução por instrução, possibilita a alteração e visualização do conteúdo das variáveis, implementa pontos de para (breakpoint) e envio de mensagem quando o conteúdo de uma variável é modificado.

1.5.7 Linguagem de controle

A linguagem de controle é a forma direta de comunicação entre o usuário e o Sistema Operacional. O usuário tem acesso às rotinas específicas do sistema.

Os comandos são interpretados pelo Shell. O Shell reconhece o comando, verifica a sintaxe, envia mensagens de erro e faz chamada de rotinas (acesso a arquivos, consulta em diretórios, etc).

A evolução da linguagem de controle são as interfaces mais amigáveis, tais como: janelas, ícones, voz, etc).