Sistemas Operacionais

Eduardo Ferreira dos Santos

Engenharia de Computação
Centro Universitário de Brasília – UniCEUB

Março, 2016

Sumário

1. Evolução
2. O Sistema Operacional
   - Gerência de processos
   - Gerência de memória
   - Gerência de arquivos
3. Sistemas Operacionais de Tempo Real

1 Evolução

2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

Histórico

- Os sistemas operacionais evoluem junto com o computador.
- Das válvulas ao Altair.
- A chegada dos periféricos e a interface com o usuário.
- A massificação do computador pessoal.
- Os primeiros casos de *vendor lock-in*.
- Linux e a revolução silenciosa.

Primeira Geração: ENIAC e computadores de grande porte

- Onde está a interface homem-máquina?

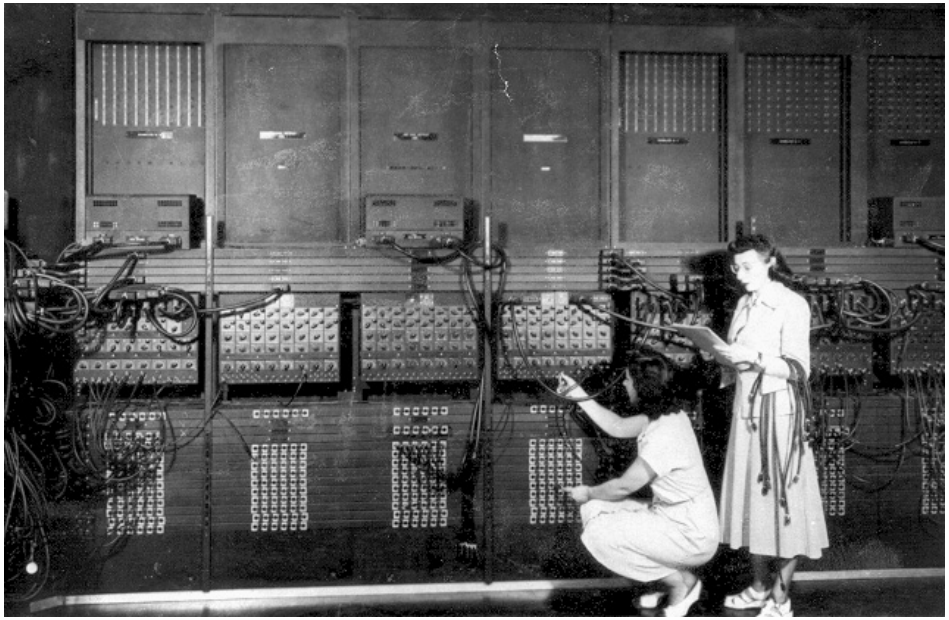


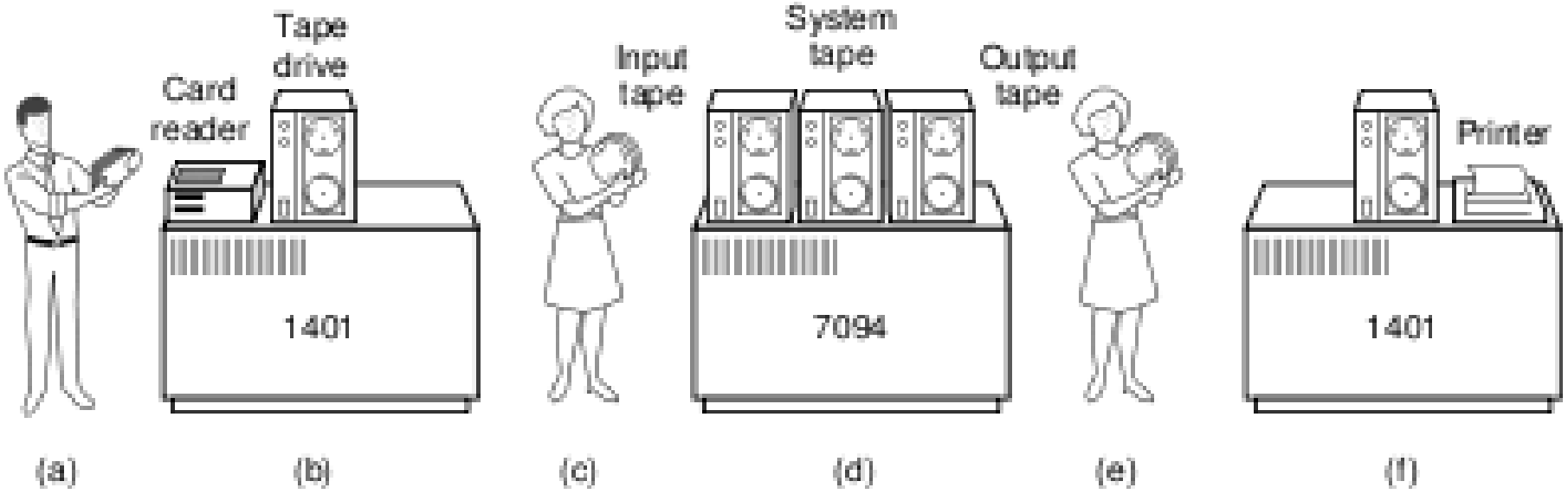
Figura 1.1: ENIAC em Operação [Penn, 2016]

Em uma análise mais profunda, podemos dizer que não havia Sistema Operacional.

Segunda Geração: Transistores e Sistemas em Lote (Batch)

- Com os transistores, os computadores passam a poder ser comercializados.
- A entrada dos dados era feita em cartões perfurados e a linguagem de máquina (assembly) foi desenvolvida para acelerar a operação.
- O processamento acontecia em lotes (batch).
- Início do que começamos a chamar de programação.

Exemplo de Processamento em Lote



[Silva, 2010]

Terceira Geração: CI's e multiprogramação

- CTSS (Compatible Time Sharing System) do MIT: primeiro sitema de tempo compartilhado;
- MULTICS (Multiplexed Information and Computing Service: MIT, Bell Labs, General Eletric;
  - Projetado para suportar centenas de usuários: uma enorme máquina (pouco mais potente que um PC) fornecendo poder computacional para toda a área de Boston;
- Unix: Ken Thompson, Bell Labs. Pai de quse todos os SO's atuais.

Quarta Geração: Computadores Pessoais

- Desenvolvimento dos circuitos integrados de larga escala (LSI, Large Scale Integration), que permitiu o surgimento dos computadores pessoais;
- 1974: Intel 8080, CPU de 8 bits de propósito geral;
- Início dos anos 80: IBM PC;
- Xerox Parc;
- Apple Lisa e Macintosh;
- MS-DOS: Microsoft, baseado no DOS, desenvolvido por Tim Paterson da Seattle Computer Products comprada por Bill Gates;
- Minix e Linux.

1 Evolução

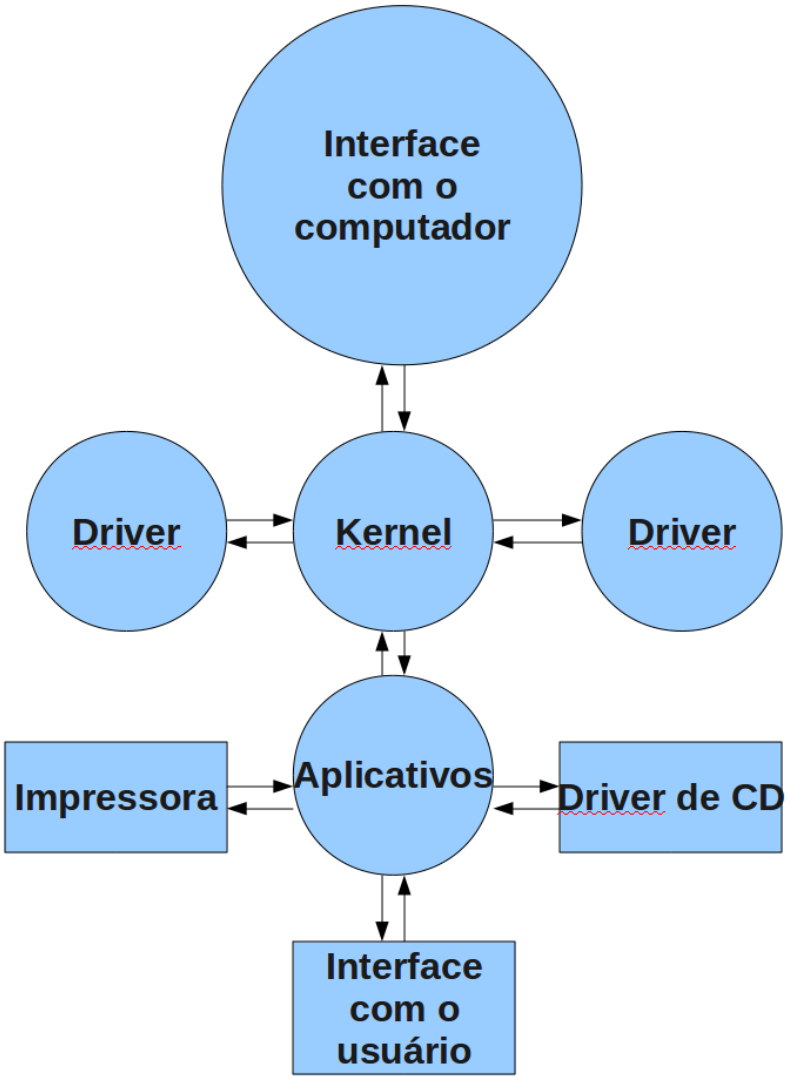
2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

O que é um Sistema Operacional?

É o programa que realiza a interface entre o Hardware e o Software.

Como funciona o Sistema Operacional



Kernel

- É o núcleo ou coração do Sistema operacional;
- Área isolada da memória;
- Kernel monolítico;
- SYSCALL;
- Através dos *device drivers*, realiza a comunicação com os periféricos e dispositivos de entrada e saída.
- O computador "diz" ao dispositivo o que quer fazer.
- O dispositivo "interpreta" através do *driver*.

1 Evolução

2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

Tipos de processamento

- Tarefas:

  Unidades de processamento sequencial que concorrem sobre um ou mais recursos computacionais de um sistema. [FARINES and MELO, 2000]

- A tarefa executada em um sistema operacional é chamada de processo.
- Processamento sequencial x Multiprogramação
- Programação concorrente:
  1. Programas separados;
  2. Threads.
- Interação em programação concorrente:
  - Memória compartilhada;
  - Troca de mensagens.

Princípios

Processos são programas que estão sendo executados em um espaço virtual de endereçamento exclusivo.

- Em sistemas Unix, processos são estruturas de dados que contém informações necessárias para a execução do programa, como conteúdo dos registradores e memória;
- Princípio básico: separar a operação de criação de um processo da operação de execução de um programa;
- São separadas por chamadas de função diferentes: *fork()* e *exec()*;
  - *fork()* cria um processo que tem como pai o processo que a chamou;
  - *exec()* cria um novo programa como uma sequência de processos, que tem como pai o contexto de execução, muitas vezes o próprio init.

Multiprogramação

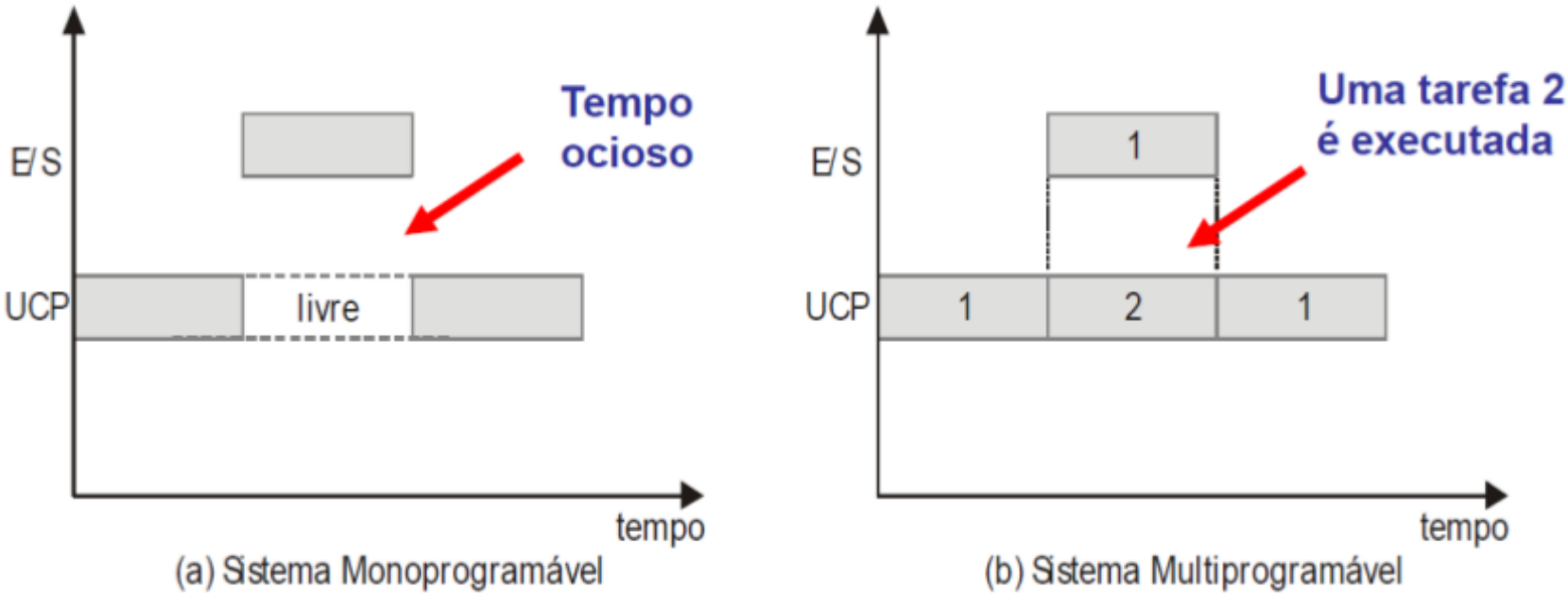


Figura 2.2: Modelo de multiprogramação [Chagas, 2016]

Estados dos processos

Durante o ciclo de vida de um processo ele passa por diferentes estados. Em sistemas Unix [Guarezi and Silva, 2010] são:

run Está sendo executado no processador;

ready ou executável Dispõe de todos os recursos que precisa e está pronto para ser executado;

sleep ou dormente Bloqueado à espera de algum recurso, e só pode ser desbloqueado se receber um sinal de outro processo;

zumbi Caso cada vez mais raro, onde um processo é criado por um programa, que por sua vez é finalizado antes de receber o resultado do processo;

parado Recebeu ordem do administrador para interromper a execução. Será reiniciado se receber um sinal de continuação (CONT).

Threads

- *Threads* no Unix são implementadas através de funções da biblioteca *libpthreads*. Existem duas formas de partilhar a CPU [Souto, 2010]:

  PTHREAD_SCOPE_PROCESS Compartilhamento somente entre *threads* do mesmo processo. Implementadas no nível do usuário;

  PTHREAD_SCOPE_SYSTEM Compartilhamento entre todas as *threads* do sistema. Implementadas no nível do kernel.

- **Curiosidade:** o Linux implementa somente *threads* em nível do usuário.

Escalonamento

- Normalmente o escalonamento é feito através um algoritmo de prioridade com *round-robin*;
- Para evitar *starvation*, cada vez que o processo é executado até o fim do quantum sua prioridade diminui;
- O grupo POSIX define três algoritmos de prioridade que podem ser utilizados em sistemas Unix:

  SCHED_FIFO FIFO preemptivo baseado em prioridades;
  SCHED_RR *Round-robin* conforme descrito;
  SCHED_OTHER Algoritmo que depende da implementação.
- **Curiosidade** - o quantum do Unix em *round-robin* é o mesmo há mais de 20 anos: 100ms [Neto, 2010].

Pipeline

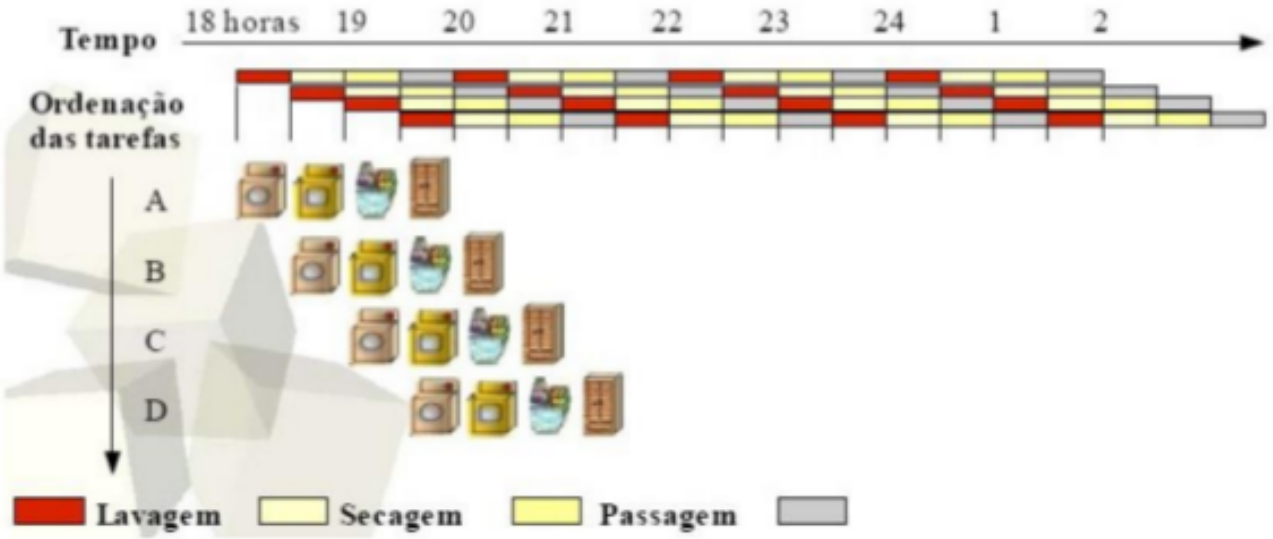


Figura 2.3: Algoritmo de pipeline [Chagas, 2016]

1 Evolução

2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

Princípios

Gerenciamento de memória consiste em manter o controle das partes da memória que estão sendo utilizadas e por quem, decidir que processos serão carregados para a memória quando houver espaço disponível, alocar e desalocar espaço quando necessário.
[Dias, 2005]

- Alocar tanto memória principal quanto memória secundária;
- Alocação da memória secundária é papel do sistema gerenciador de arquivos.

Memória virtual e paginação

- Sistemas Unix utilizam memória virtual e paginação;
- Estratégia de memória virtual armazena no disco partes do programa que não estão sendo utilizadas (swapping);
- O espaço de endereçamento virtual é divido em unidades de tamanho fixo chamadas páginas;
- A conversão para um endereço físico é feita por um componente de *hardware* chamado MMU;
- O controle da MMU é feito pelo *driver* compilado com o kernel.

Divisão da memória principal

- A memória principal é divida em alguns segmentos com função específica:

  kernel Parte da memória reservada exclusivamente ao kernel de acesso restrito;

  buffer do disco Parte que armazena os arquivos que serão enviados ao disco. O controle de escrita é feito pelo kernel, mas os arquivos só são enviados para o disco quando o *buffer* está cheio;

  descritores de arquivo Contém partes das tabelas de descrição dos arquivos, para agilizar o acesso;

  programas O resto da memória é destinado aos programas.

- Importante: definir o tamanho de cada um dos espaços reservados na memória — exceção feita ao kernel — afeta bastante o desempenho do sistema operacional.

1 Evolução

2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

O gerenciador de arquivos

- A memória, tanto principal quanto secundária, também é responsabilidade do Sistema Operacional.
- Quando salvamos o arquivo no disco, estamos executando as seguintes tarefas:
  1. O programa pede ao sistema operacional para escrever o conteúdo de uma arquivo;
  2. O sistema operacional repassa a tarefa para o gerenciador de arquivos (file manager), que é um subconjunto do SO;
  3. O gerenciador de arquivos busca em uma tabela informações sobre o arquivo;

O gerenciador de arquivos

- Quando salvamos o arquivo no disco, estamos executando as seguintes tarefas (continuação):
  1. O gerenciador de arquivos busca em uma tabela a localização física do setor que deve conter o byte (cilindro, trilha, setor);
  2. O gerenciador de arquivos instrui o processador de I/O (que libera a CPU de cuidar do processo de transferência) sobre a posição do byte na RAM, e onde ele deve ser colocado no disco;
  3. O processador de I/O formata o dado apropriadamente, e decide o melhor momento de escrevê-lo no disco.

Princípios

- Os discos são vistos pelo SO como uma série de blocos de tamanho fixo;
- A alocação no disco não é realizada diretamente pelo SO;
- Uma chamada do sistema (*SYSCALL*) chama o *driver* do dispositivo que é compilado junto com o kernel;
- O *driver* informa a quantidade de espaço disponível no disco.

Paradigma da computação: gargalo de Von Neumann

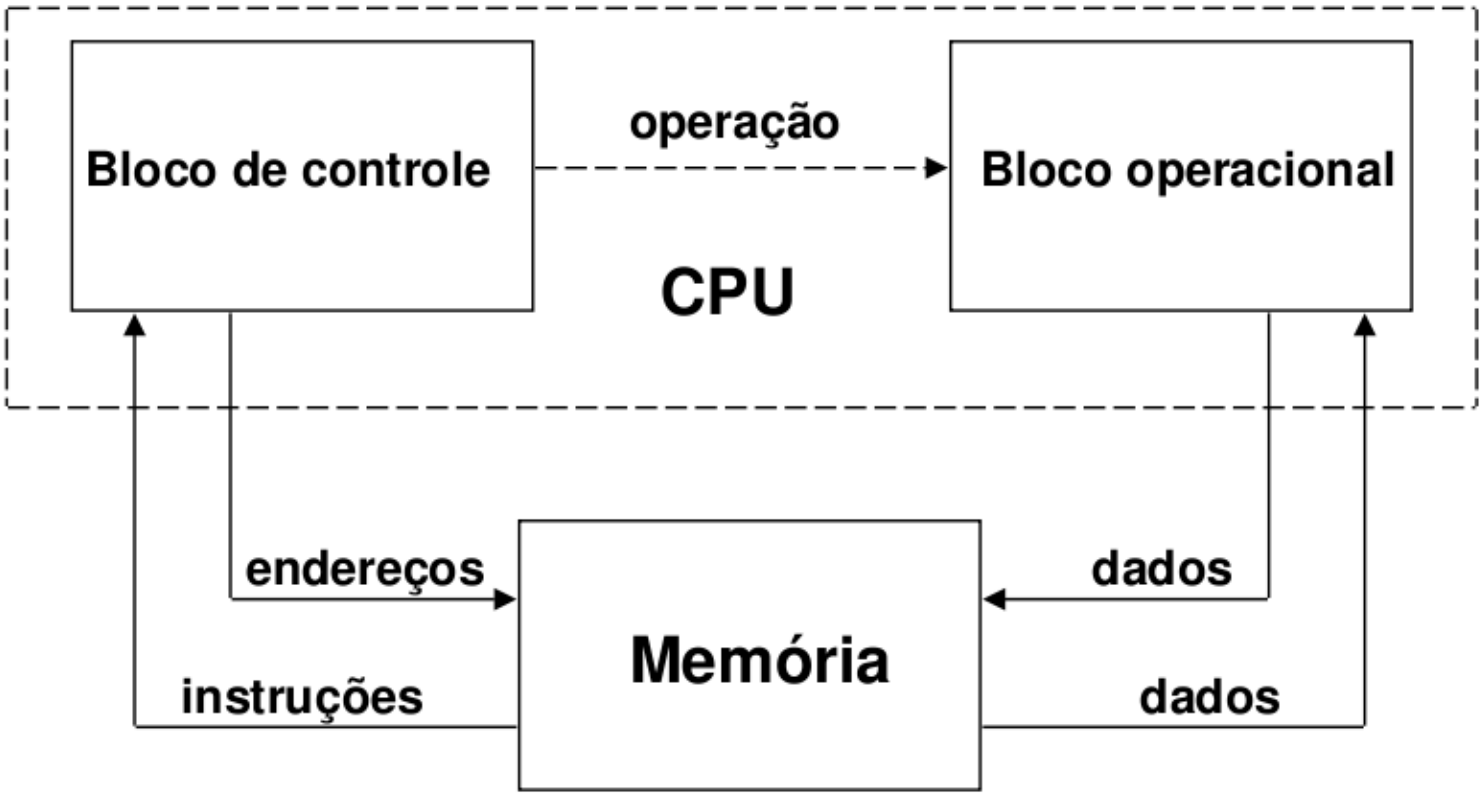


Figura 2.4: Abordagem de Von Neumann

Qual o problema da abordagem de Von Neumann?

Unix como base

- O Unix é a base para quase tudo o que conhecemos como Sistema Operacional;
- Conhecer seu funcionamento é análogo a conhecer bem o computador;
- Sua grande contribuição só foi possível por causa de suas bases abertas, uma grande coincidência histórica;
- Grande questão do momento: desenvolvimento colaborativo (família BSD) ou sistemas comerciais (Apple, IBM, Oracle/Sun)?

1 Evolução

2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

O problema tempo real

Verificar e implementar sistemas ou programas que, mesmo com recursos limitados, apresentam comportamentos previsíveis, atendendo as restrições temporais impostas pelo ambiente ou pelo usuário. [FARINES and MELO, 2000]

- As tarefas de tempo real estão atreladas a seus prazos: deadlines
- Importância da troca de mensagens;
- Problemas do escalonamento por deadline:
  - *deadlocks*
  - *polling*
  - *Release jitter*

Escalonamento de tempo real

- O foco de um sistema operacional de tempo real está relacionado aos algoritmos de escalonamento;

  Carga estática O tempo de resposta de todas as tarefas é conhecido, considerando o ipor o caso;

  Cargas dinâmicas As características de chegada das tarefas não são conhecidas.

- Abordagens de escalonamento de tempo real:

  off-line guarantee Garantia em tempo de projeto;

  on-line guarantee Garantia em tempo de execução;

  best-effort Abordagem de melhor esforço.

Escalonamento de tempo real

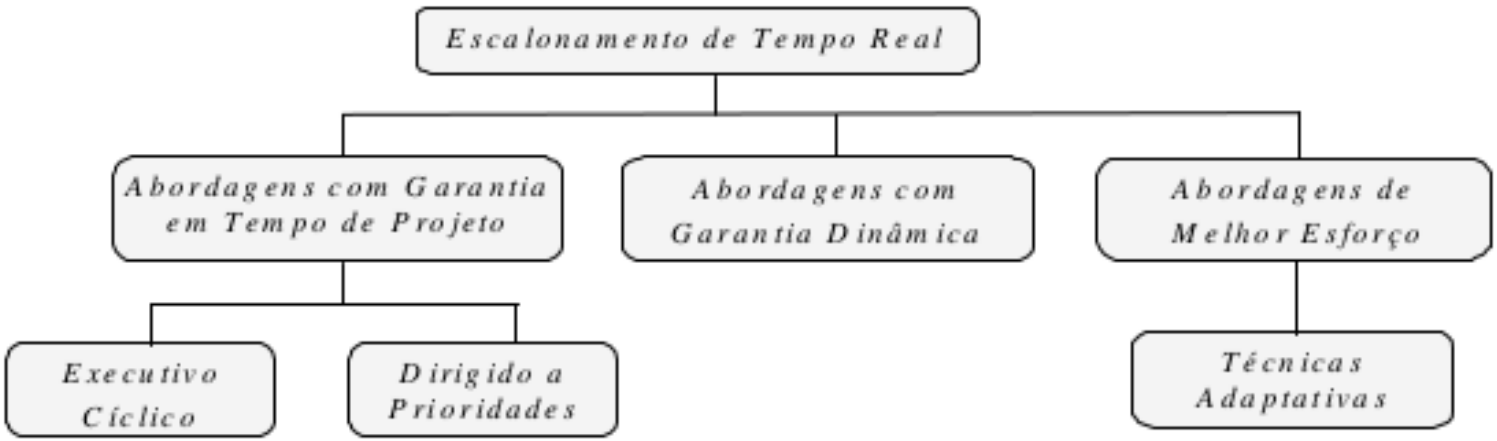


Figura 3.1: Resumo das abordagens de escalonamento de Tempo Real [FARINES and MELO, 2000, p.20]

1 Evolução

2 O Sistema Operacional
- Gerência de processos
- Gerência de memória
- Gerência de arquivos

3 Sistemas Operacionais de Tempo Real

Resumo

- Sistemas operacionais fazem a interface entre o hardware e o software;
- Processos são tarefas executadas pela CPU;
- Threads são processos que possuem área de memória compartilhada;
- Os processos e/ou threads se comunicam através de mensagens ou memória;
- Abordagens de escalonamento em SO's convencionais são orientadas a prioriedades;
- Os algorimos de escalonamento em tempo real são baseados em *deadlines*.


Chagas, F. (2016).
Notas de aula do prof. fernando chagas.


Dias, A. M. (2005).
Noções de sistemas operacionais.
Disponível em: http://www.dca.ufrn.br/~xamd/dca0800/Cap03.pdf
Acessado em 06/01/2011.


FARINES, J. M. and MELO, R. (2000).
Sistemas de Tempo Real, volume 1.
IME-USP.


Guarezi, D. J. and Silva, E. B. (2010).
Processos em windows e unix.
Disponível em:
http://www.inf.ufsc.br/~magro/PROCESSOS%20EM%20WINDOWS%2
Acessado em 28/01/2011.


Neto, D. O. G. (2010).
Processos no unix.

Disponível em:
http://homepages.dcc.ufmg.br/~dorgival/slides/so/04b-processosUnix-6pp.pdf Acessado em 28/01/2011.


Penn (2016).
Foto do eniac.


Silva, F. J. d. (2010).
Histórico do sistemas operacionais.
http://www.deinf.ufma.br/ fssilva/graduacao/so/aulas/historico.pdf Acessado em 08/06/2010.


Souto, P. F. (2010).
Sistemas operativos: Escalonamento de processos.
Disponível em: paginas.fe.up.pt/~pfs/aulas/aso0708/at/at6.pdf Acessado em 28/01/2011.

OBRIGADO!!!
PERGUNTAS???