

# SISTEMAS OPERACIONAIS

Turma de Redes

**AULA 03**

**Prof.: Silvestri**  
**[www.eduardosilvestri.com.br](http://www.eduardosilvestri.com.br)**  
**[silvestri@eduardosilvestri.com.br](mailto:silvestri@eduardosilvestri.com.br)**

Prof.: Silvestri

# Como a Computação Funcionava

ANTES:

Entrada – Chaves

Saída – Luzes

DEPOIS:

Entrada – Teclado

Saída - Monitor

## **Como a Computação Funcionava**

Programador também era o operador;

Utilização restrita;

Todo o processo de montagem de um programa para ser executado era muito lento.

Para solucionar esse problema foram criados:

- Montadores (assemblers);
- Carregadores (loaders); e
- Ligadores (linkers)

## **Como foi modificada**

As linguagens de alto nível foram criadas:

- Fortran; e
- Cobol.

## **Como foi modificada**

### O Monitor Residente

Como o tempo para preparação de Job (tarefa) era muito alto, a UCP (Unidade Central de Processamento) ficava ociosa;  
Lembrando que os computadores eram extremamente caros e o tempo de processamento tinha muito valor, e quanto mais tempo em uso melhor seria.

## Como foi modificada

### O Monitor Residente

Para reduzir o tempo, jobs com necessidades similares eram agrupados (batched) e executados em grupo pelo computador. Essa abordagem marcou a época do **Processamento em Batch**

Mesmo assim toda vez que um problema era detectado o sistema deveria ser reiniciado e um novo cartão com um programa deveria ser carregado.

## **Como foi modificada**

### O Monitor Residente

Para resolver essas inicializações que eram executadas toda hora foi criado um seqüenciador de jobs. (Primeiro Sistema Operacional).

Sua função era controlar a transferência de um job para outro. Ele funcionava como um monitor residente, estava sempre presente na memória do computador.

## Como foi modificada

O Monitor Residente

Comando:

\$JOB - Primeiro cartão, indicando o início de um *job*;  
\$FTN - Executar o compilador FORTRAN;  
\$LOAD - Carregar o programa compilado;  
\$RUN - Executar o programa carregado;  
\$END - Fim do *job*.

## **Como foi modificada**

### Bufferização

Enquanto o sistema ficava lendo as informações da fita o UCP ficava esperando, porém a idéia da bufferização fazia com que paralelamente enquanto a UCP executava uma seqüência de comandos lidos o sistema de E/S já fazia novas leituras na fita.

## Como foi modificada

### Spool

Com o passar do tempo novos dispositivos de armazenamentos foram criados DISCO que permitiam uma escrita ou leitura a qualquer momento. Diferente das fitas que para escrever era preciso avançar as fitas ate o final. Com isso todas as operações de escrita e leitura poderiam ser bufferizadas em discos e escritas a qualquer momento.

Spool - *Simultaneous Peripheral Operation On-Line*

## Como foi modificada

### Multiprogramação

Inicialmente o sistema operacional escolhe um dos *jobs* da lista de *jobs* e começa a executá-lo. Eventualmente, o *job* deve esperar por alguma tarefa, como a montagem de uma fita, um comando digitado pelo teclado, ou mesmo o término de uma operação de E/S. Em um sistema **monoprogramado** a UCP permaneceria ociosa.

## Como foi modificada

### Multiprogramação

Para que vários *jobs* estejam prontos para executar, é necessário que todos estejam presentes na memória RAM da máquina simultaneamente. Isto acarreta em um gerenciamento de memória para os vários *jobs*. Além disso, se vários *jobs* estão prontos para executar ao mesmo tempo, o sistema deve escolher qual deles deve ser executado primeiro. A política de decisão de qual *job* será executado é chamada de **escalonamento** de UCP.

## Como foi modificada

### Multiprogramação

Por fim, o sistema operacional deve garantir que vários *jobs* rodando concorrentemente não afetem uns aos outros em todas as fases do sistema operacional, incluindo escalonamento de processos, armazenamento de disco e gerenciamento de memória.

## Como foi modificada

### Tempo Compartilhado

O conceito de sistemas de **tempo compartilhado**, também chamados de **multitarefa**, é uma extensão lógica de multiprogramação.

## Como foi modificada

### Tempo Compartilhado

Neste ambiente, múltiplos *jobs* são executados simultaneamente, sendo que a UCP atende cada *job* por um pequeno tempo, um a um em seqüência. Os tempos dedicados para cada *job* são pequenos o suficiente para que os usuários consigam interagir com cada programa sem que percebam que existem outros programas rodando.

## Como foi modificada

### Tempo Compartilhado

Sistemas de tempo compartilhado foram desenvolvidos para fornecer o uso interativo de um sistema de computação a custos razoáveis. Um **sistema operacional de tempo compartilhado** (*time-sharing*) usa **escalonamento de UCP** e **multiprogramação** para fornecer a cada usuário uma pequena porção de tempo de computador.



***DÚVIDAS***

Prof.: Silvestri

# ***Perguntas***

- 1 – Como as coisas funcionavam antes ?***
- 2 – O que fazia o primeiro SO ?***
- 3 – Como ele funcionava ?***
- 4 – O que é processamento em Batch ?***
- 5 – Como funciona a bufferização ?***
- 6 – Como funciona o Spool ?***
- 7 – Como funciona a Multiprogramação ?***
- 8 – O que é Escalonamento ?***
- 9 – Como funciona o sistema por Tempo Compartilhado ?***
- 10 – O que é Escalonamento ?***

## **Publicação.**

- As questões respondidas; e
- Mapa conceitual.

## **Entrega.**

- Mapa conceitual.