

Sistemas Operacionais

Gestão de tarefas - o conceito de tarefa

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Fevereiro de 2019

Conteúdo

- 1 Programas e tarefas
- 2 Sistemas monotarefa
- 3 Sistemas multitarefa
- 4 Tempo compartilhado
- 5 Estados e transições

Objetivos

Questões a serem respondidas:

- Como as tarefas são definidas
- Quais os estados possíveis de uma tarefa
- Como/quando o processador troca de tarefa
- Como ordenar as tarefas para usar o processador

Programas e tarefas

Programa

- sequência de instruções para resolver um problema específico
- são aplicações ou utilitários
- define um conceito **estático**, sem estado interno

Exemplos:

- `C:\Windows\notepad.exe`
- `/usr/bin/firefox`

Programas e tarefas

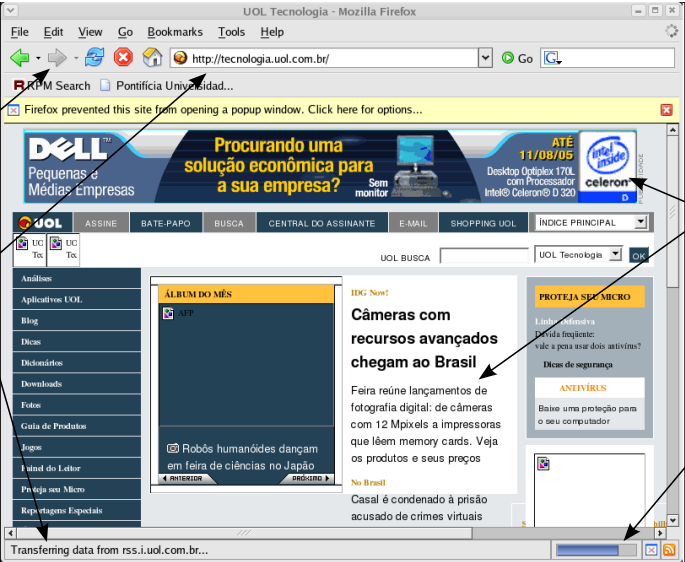
Tarefa:

- execução, pelo processador, das sequências de instruções definidas em um programa
- conceito **dinâmico**, com estado interno
- Estado interno evolui a cada instante
- Implementada de várias formas (processos, *threads*)

Exemplos:

- O *Notepad* editando um arquivo `readme.txt`
- O visualizador de PDFs mostrando este slide

Tarefas de um navegador



The screenshot shows the Mozilla Firefox browser interface with the following elements and annotations:

- 4**: Points to the navigation buttons (back, forward, home, stop, refresh) in the browser toolbar.
- 1**: Points to the address bar and search engines (RPM Search, Pontificia Univer...
- 2**: Points to the main content area, specifically the article titled "Câmeras com recursos avançados chegam ao Brasil".
- 3**: Points to the status bar at the bottom, which shows "Transferring data from rss.i.uol.com.br...".

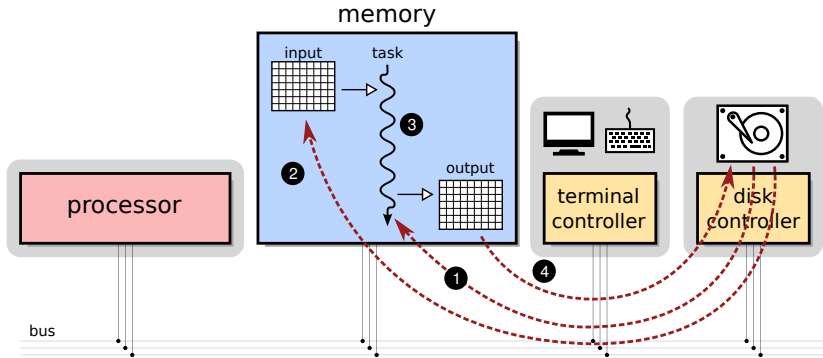
Sistemas monotarefa

Primeiros sistemas de computação.

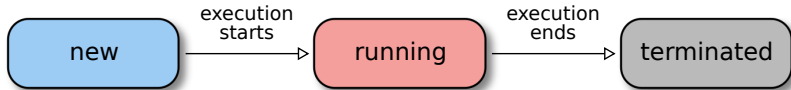
Executam uma única tarefa de cada vez:

- 1 o programa binário é carregado do disco para a memória
- 2 os dados do programa são carregado na memória
- 3 o programa executa até sua conclusão
- 4 os resultados do programa são descarregados da memória

Sistemas monotarefa



Estados de uma tarefa



Sistemas multitarefa

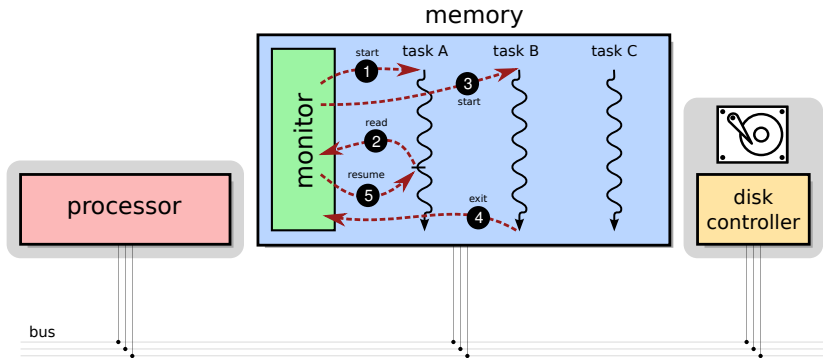
Problemas:

- Tarefas esperando por entrada/saída ficam paradas
- Custo de operação do computador era muito elevado
- UNIVAC I: 125 kW (1951)

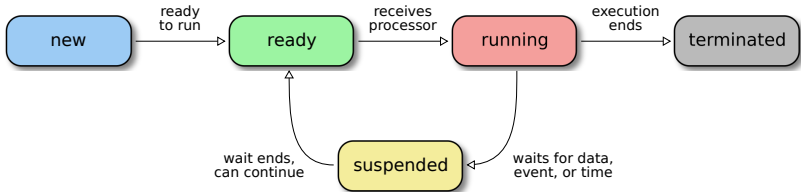
Solução:

- Usar o processador ocioso para tratar outras tarefas
- Carregar várias tarefas na memória
- Um software *monitor* coordena a troca de tarefas

Sistemas multitarefa



Estados de uma tarefa



Sistemas de tempo compartilhado

Problemas:

- aplicações em laço infinito podem bloquear o sistema
- melhorar o comportamento de aplicações interativas

```
1 void main ()
2 {
3     int i = 0, soma = 0 ;
4
5     while (i < 1000)
6         soma += i ; // erro: i não foi incrementado
7
8     printf ("A soma vale %d\n", soma);
9 }
```

Sistemas de tempo compartilhado

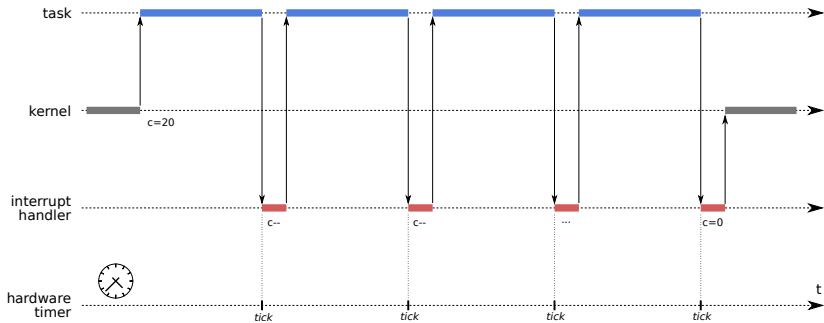
Solução: *Time Sharing*

- Tarefas recebe fatias de tempo (quantum) de CPU
- *Quantum* típico vai de 10 ms a 200 ms
- Implementado através de interrupções de tempo (*ticks*)

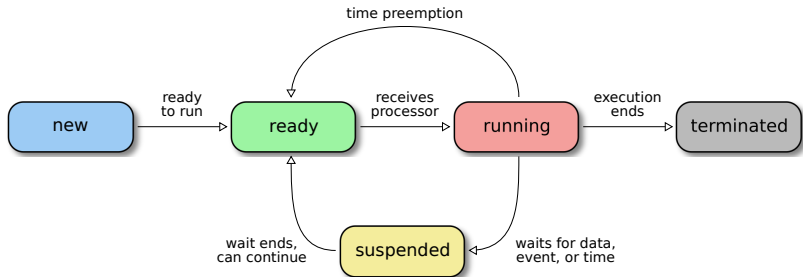
Passos:

- 1 Tarefa recebe o processador
- 2 Tarefa executa até expirar seu *quantum*
- 3 Tarefa retorna à fila de espera pela CPU
- 4 Outra tarefa recebe o processador

Sistemas de tempo compartilhado



Sistemas de tempo compartilhado



Estados das tarefas

Nova : A tarefa está sendo preparada para executar

Pronta : A tarefa está esperando o processador

Executando : O processador está executando a tarefa

Suspensa : A tarefa aguarda algum evento externo

Terminada : A tarefa foi encerrada

Transições das tarefas

- $\dots \rightarrow N$ a tarefa ingressa no sistema
- $N \rightarrow P$ a tarefa está pronta para executar
- $P \rightarrow E$ a tarefa é escolhida para executar
- $E \rightarrow P$ esgota a fatia de tempo da tarefa
- $E \rightarrow T$ a tarefa encerra sua execução
- $T \rightarrow \dots$ a tarefa terminada é removida da memória
- $E \rightarrow S$ a tarefa em execução decide aguardar um recurso ou evento externo
- $S \rightarrow P$ o recurso ou evento aguardado pela tarefa está disponível

Tarefas no Linux

Comando top:

```

1 top - 16:58:06 up 8:26, 1 user, load average: 6,04, 2,36, 1,08
2 Tarefas: 218 total, 7 executando, 211 dormindo, 0 parado, 0 zumbi
3 %Cpu(s): 49,7 us, 47,0 sy, 0,0 ni, 3,2 id, 0,0 wa, 0,0 hi, 0,1 si, 0,0 st
4 KiB Mem : 16095364 total, 9856576 free, 3134380 used, 3104408 buff/cache
5 KiB Swap: 0 total, 0 free, 0 used. 11858380 avail Mem
6
7 PID USUÁRIO PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
8 32703 maziero 20 0 2132220 432628 139312 S 44,8 2,7 0:53.64 Web Content
9 2192 maziero 20 0 9617080 686444 248996 S 29,8 4,3 20:01.81 firefox
10 11650 maziero 20 0 2003888 327036 129164 R 24,0 2,0 1:16.70 Web Content
11 9844 maziero 20 0 2130164 442520 149508 R 17,9 2,7 1:29.18 Web Content
12 11884 maziero 20 0 25276 7692 3300 S 15,5 0,0 0:37.18 bash
13 20425 maziero 20 0 24808 7144 3212 S 14,4 0,0 0:08.39 bash
14 1782 maziero 20 0 1788328 235200 77268 S 8,7 1,5 24:12.75 gnome-shell
15 ...
  
```