

Fundamentos de Processamento Paralelo e Distribuído

Sistemas Sequenciais

lembrando conceitos básicos

Fernando Luís Dotti

Processo Sequencial

- Processo: programa em execução
- Processo sequencial: programa sequencial em execução
- Ciclo de vida de vida de processos
- Noções de sistemas operacionais

Processo Sequencial

- Ciclo de vida
 - carga, execução, término
- Execução
 - ciclo de instruções
 - busca da memória
 - executa instrução:
 - armazena na memória, em registrador, jump, etc
 - até acabar...
- Máquina de Turing

Carga

Representação

```
Fib(n){  
  var vetor f [1..n] de inteiros  
  início  
    f[1]=0  
    f[2]=1  
    para i=3 ate n  
      f[i] = f[i-2]+f[i-1]  
    escreve f na tela  
  fim
```

Carga Execução

Representação

```
Fib(n){  
  var vetor f [1..n] de inteiros  
  início  
    f[1]=0  
    f[2]=1  
    para i=3 ate n  
      f[i] = f[i-2]+f[i-1]  
    escreve f na tela  
  fim
```

Representação

Carga
Execução
Término

```
Fib(n){  
  var vetor f [1..n] de inteiros  
  início  
    f[1]=0  
    f[2]=1  
    para i=3 ate n  
      f[i] = f[i-2]+f[i-1]  
    escreve f na tela  
  fim
```

Processo Sequencial

Carga

aloca memória para
dados e instruções

Fib(n){

var vetor f [1..n] de inteiros

início

f[1]=0

f[2]=1

para i=3 ate n

f[i] = f[i-2]+f[i-1]

escreve f na tela

fim

Processo Sequencial

Carga

aloca memória para
dados e instruções

Execução

a partir do estado inicial PC=0
fetch e executa instrução,
mudando o estado

Fib(n){

var vetor f [1..n] de inteiros

início

→ f[1]=0

· f[2]=1

· para i=3 ate n

· f[i] = f[i-2]+f[i-1]

· escreve f na tela

fim

Resultando uma sequência de estados

Processo Sequencial

estado
inicial

$f = [\perp, \perp, \dots]$
 $i = \perp \quad n=5$
pc = 0

Fib(n){

var vetor f [1..n] de inteiros

início

0 -> $f[1]=0$

1 -> $f[2]=1$

2 -> para i=3 ate n

3 -> $f[i] = f[i-2]+f[i-1]$

4 -> escreve f na tela

5 -> **fim**

Processo Sequencial

estado
inicial

$f = [\perp, \perp, \dots]$
 $i = \perp \quad n=5$
pc = 0

$f = [0, \perp, \dots]$
 $i = \perp \quad n=5$
pc = 1

fetch,
exec

Fib(n){

var vetor f [1..n] de inteiros

início

0 -> $f[1]=0$

1 -> $f[2]=1$

2 -> para $i=3$ ate n

3 -> $f[i] = f[i-2]+f[i-1]$

4 -> escreve f na tela

5 -> fim

Processo Sequencial

estado
inicial

$f = [\perp, \perp, \dots]$
 $i = \perp \quad n=5$
pc = 0

fetch,
exec

$f = [0, \perp, \dots]$
 $i = \perp \quad n=5$
pc = 1

fetch,
exec

$f = [0, 1, \dots]$
 $i = \perp \quad n=5$
pc = 2

Fib(n){

var vetor f [1..n] de inteiros

início

0 -> $f[1]=0$

1 -> $f[2]=1$

2 -> para i=3 ate n

3 -> $f[i] = f[i-2]+f[i-1]$

4 -> escreve f na tela

5 -> fim

Processo Sequencial

estado
inicial

Fib(n){

var vetor f [1..n] de inteiros

início

0 -> f[1]=0

1 -> f[2]=1

2 -> para i=3 ate n

3 -> f[i] = f[i-2]+f[i-1]

4 -> escreve f na tela

5 -> fim

FIM

f = [0, 1, 1, 2, 3]
i = 5 n=5
pc = 5

...

f = [0, 1, 1, 2, 3]
i = 5 n=5
pc = 4

f = [0, 1, 1, 2, 3]
i = 5 n=5
pc = 2

fetch, exec

f = [0, 1, 1, 2...]
i = 5 n=5
pc = 3

fetch, exec

f = [0, 1, 1, 2...]
i = 4 n=5
pc = 2

fetch, exec

f = [0, 1, 1 ...]
i = 4 n=5
pc = 3

fetch, exec

f = [0, 1, 1 ...]
i = 3 n=5
pc = 2

fetch,
exec

f = [0, 1, ...]
i = 3 n=5
pc = 3

fetch,
exec

f = [0, 1, ...]
i = 1 n=5
pc = 2

fetch,
exec

f = [0, 1, ...]
i = 1 n=5
pc = 1

fetch,
exec

f = [1, 1, ...]
i = 1 n=5
pc = 0

