Laboratório de Computadores: Processos

2º MIEIC

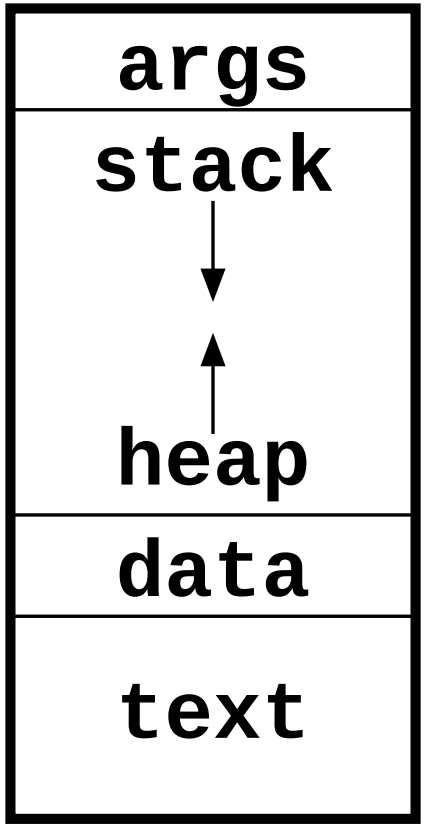
Pedro F. Souto (pfs@fe.up.pt)

October 27, 2010

Processo (Sequencial)

Representa um programa em execução

```
int main(int argc, char *argv[], char* envp[])}
```



`args` Argumentos passados na linha de comando e variáveis de ambiente.

`stack` *Registos de activação* correspondente à invocação de funções.

`heap` Dados alocados dinamicamente usando `malloc`.

`data` Dados alocados estaticamente pelo compilador (p.ex. a *string* "`Hello, World!`")

`text` Instruções do programa.

Unix/Linux são SOs multiprocesso (XP,Vista,...)

```
i02cu0108:~> ps ax | more
  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:05 init
    2 ?        SW     0:00 [kflushd]
    3 ?        SW     0:03 [kupdate]
    4 ?        SW     0:00 [kpiod]
    5 ?        SW     0:00 [kswapd]
   82 ?        S      0:00 /sbin/portmap
  139 ?        S      0:00 /sbin/syslogd
  141 ?        S      0:00 /sbin/klogd
  147 ?        S      0:00 /sbin/rpc.statd
  149 ?        SW     0:00 [lockd]
  150 ?        SW     0:00 [rpciod]
  158 ?        S      0:00 /usr/sbin/ypbind
--more (63)
```

SOs suportam múltiplos processos (multiprogramação) por razões de **eficiência**

Multi-processo e eficiência

Problema processos precisam de aceder a dispositivos periféricos de entrada e saída de dados (consola, i.e. o monitor e o teclado, rato, disco, modem, placa de rede, etc):

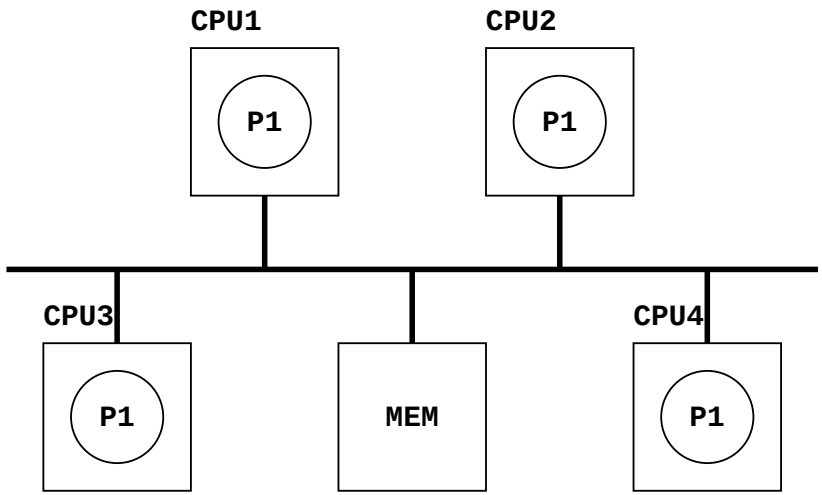
Parâmetro	Tempo
Ciclo do CPU	1 ns (1 GHz)
Acesso à *cache*	~ 2ns
Acesso à memória	~10 ns
Acesso ao disco	~10 ms

Solução quando um processo inicia uma operação de entrada/saída de dados e fica à espera que ela termine, o sistema operativo atribui o processador a outro processo:

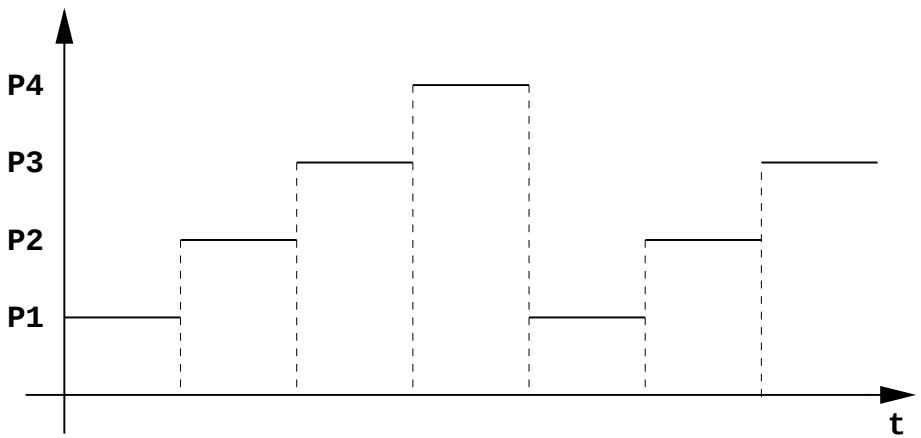
- quando a operação terminar o periférico gera uma interrupção

Execução multi-processo (1/2)

- Em sistemas multiprocessador (i), vários processos podem executar ao mesmo tempo: um em cada processador.
- Num sistema uniprocessador (ii), o sistema operativo gere a atribuição do processador aos diferentes processos (o processador é um recurso partilhado pelos diferentes processos): *pseudo-paralelismo*.

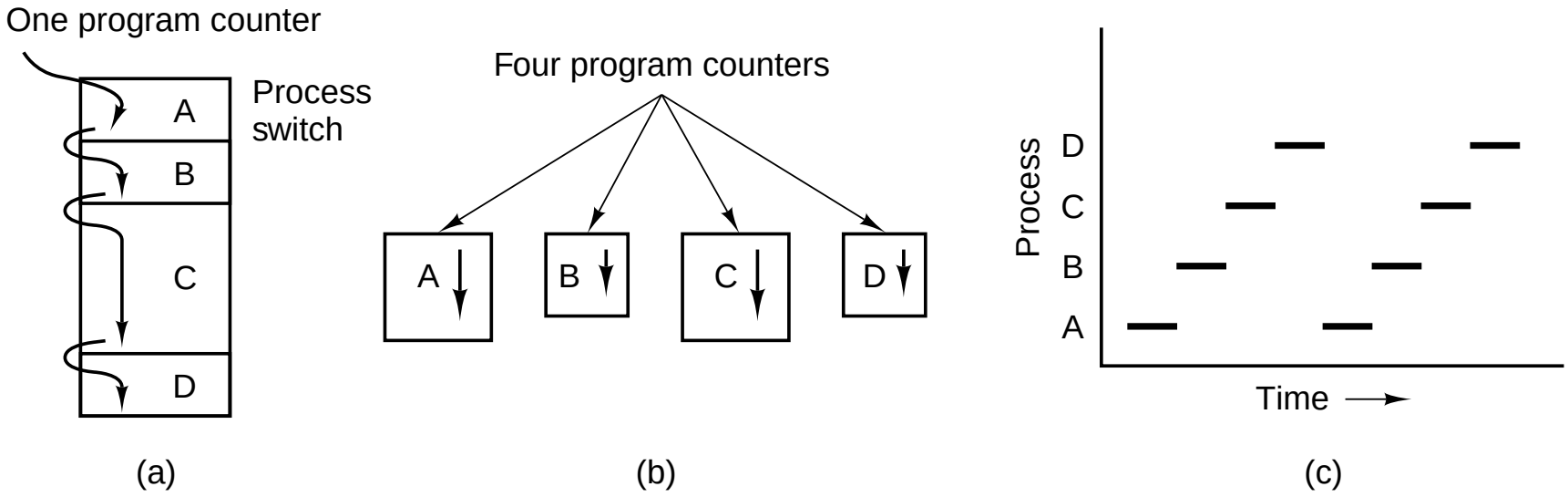


(i)



(ii)

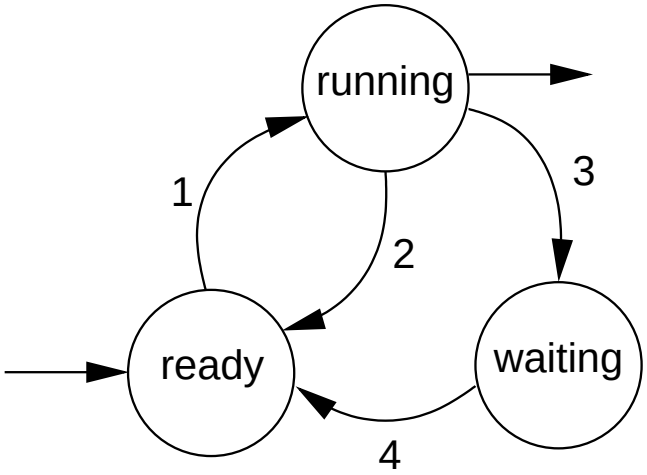
Execução multi-processo (2/2)



- O computador é partilhado por 4 processos;
- O SO dá a ilusão de que cada processo executa isoladamente num CPU, i.e. cada processo executa num CPU virtual.

Estados dum Processo

- Ao longo da sua existência um processo pode estar em 1 de 3 estados:



1. CPU atribuído ao processo (pelo SO);
2. CPU removido do processo (pelo SO);
3. processo bloqueia à espera dum evento;
4. ocorrência do evento esperado.

execução *(running)*: o CPU está a executar as instruções do processo;

bloqueado *(waiting)*: o processo está à espera de um evento externo (tipicamente, o fim de uma operação de E/S) para poder prosseguir;

pronto *(ready)*: o processo está à espera do CPU, o qual está a executar instruções de outro processo.

Further Reading

- Secções 2, 2.1
  Andrew Tanenbaum, *Modern Operating Systems*, 2nd Ed.