



UNIFOR
ENSINANDO E APRENDENDO

Disciplina de Sistemas Operacionais (Aula 2)



Introdução ao Shell Script

Apresentação



Francisco Nauber Bernardo Gois

Analista aprendizado de máquina
no Serviço Federal de Processamento
de Dados

Doutorando em Informática Aplicada

Mestre em Informática Aplicada

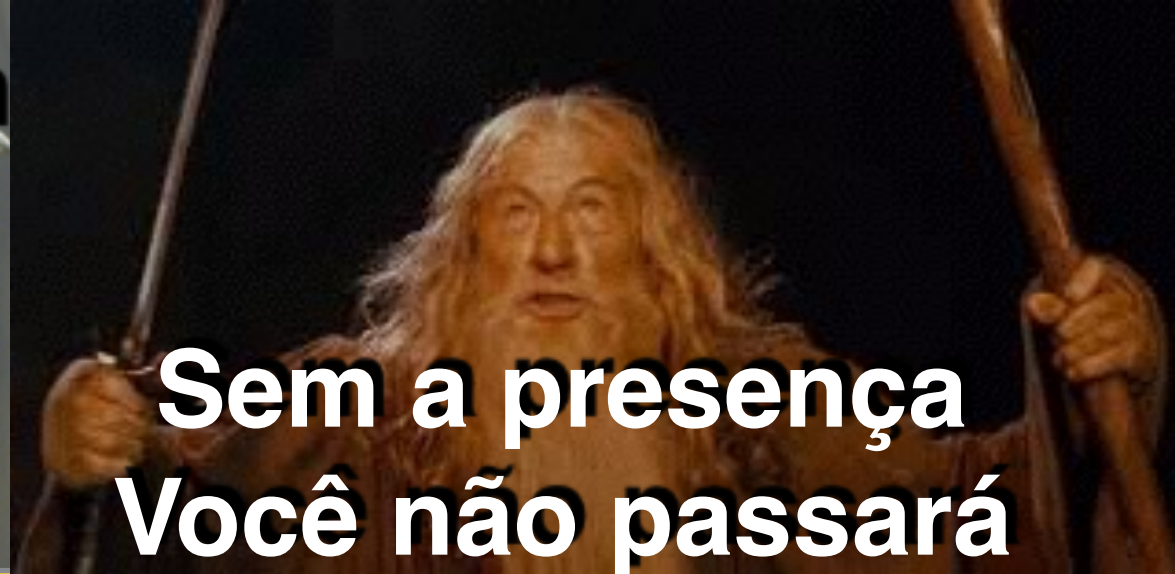
Especialista em desenvolvimento WEB

**Jovem Padawan
procure na aula**



**ao telefone
não falar**

**Sem a presença
Você não passará**



**Para melhor
desempenho na
aula**

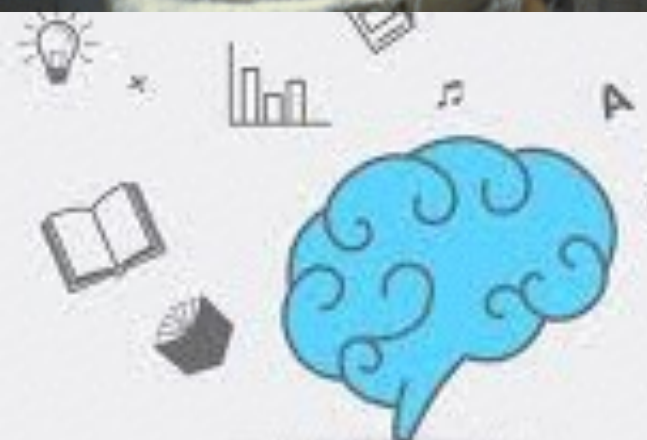
**Cuidado
com o
Horário**

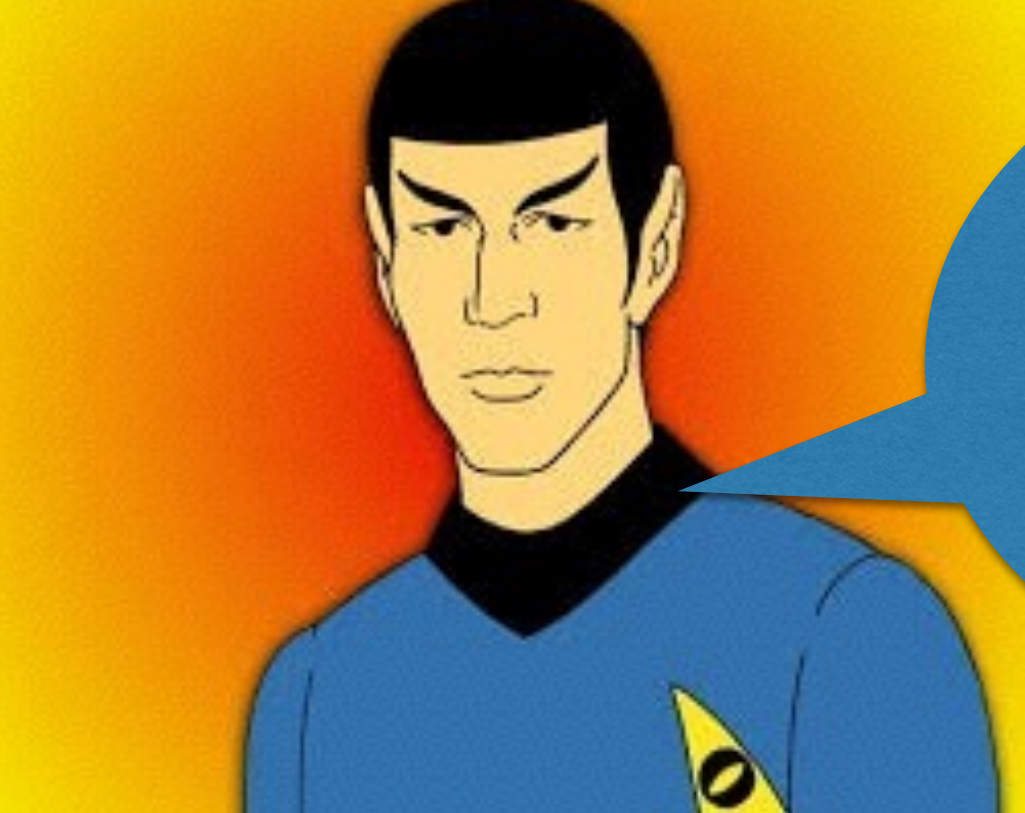
**Não
teremos
pontuação fora dos
trabalhos e provas
da disciplina**

**Buscar
aprendizado
ao invés de
pontos**



**Procure
não
conversar
durante a
aula**





OS trabalhos
deveram ser
entregues uma
semana antes
da prova

Um cadeira longa
e prospera

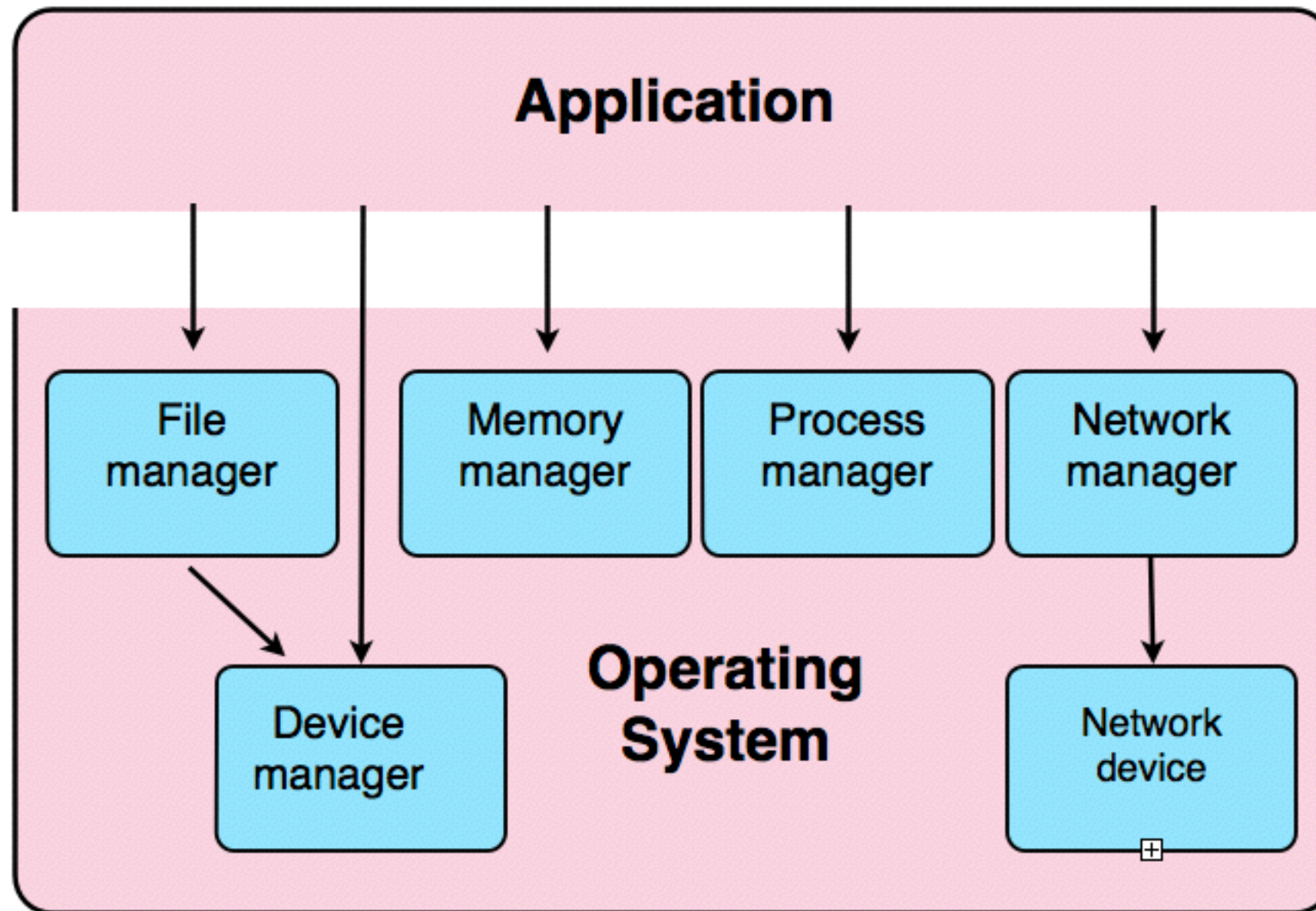
Não teremos
pontos após a prova
não adianta pedir



Cronograma da Disciplina



Gerências de recursos dos Sistemas Operacionais



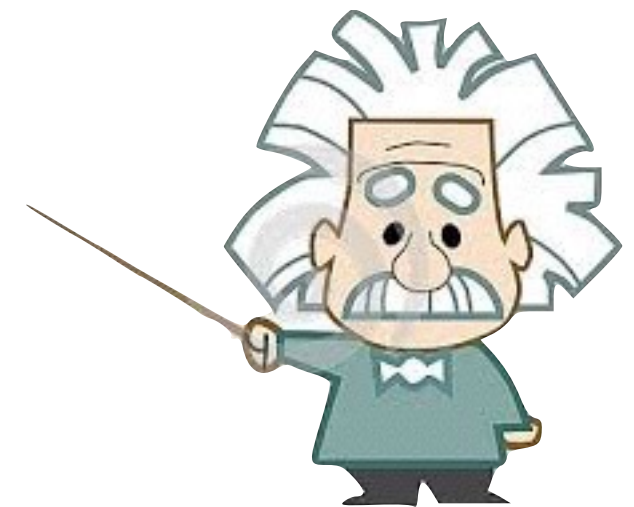
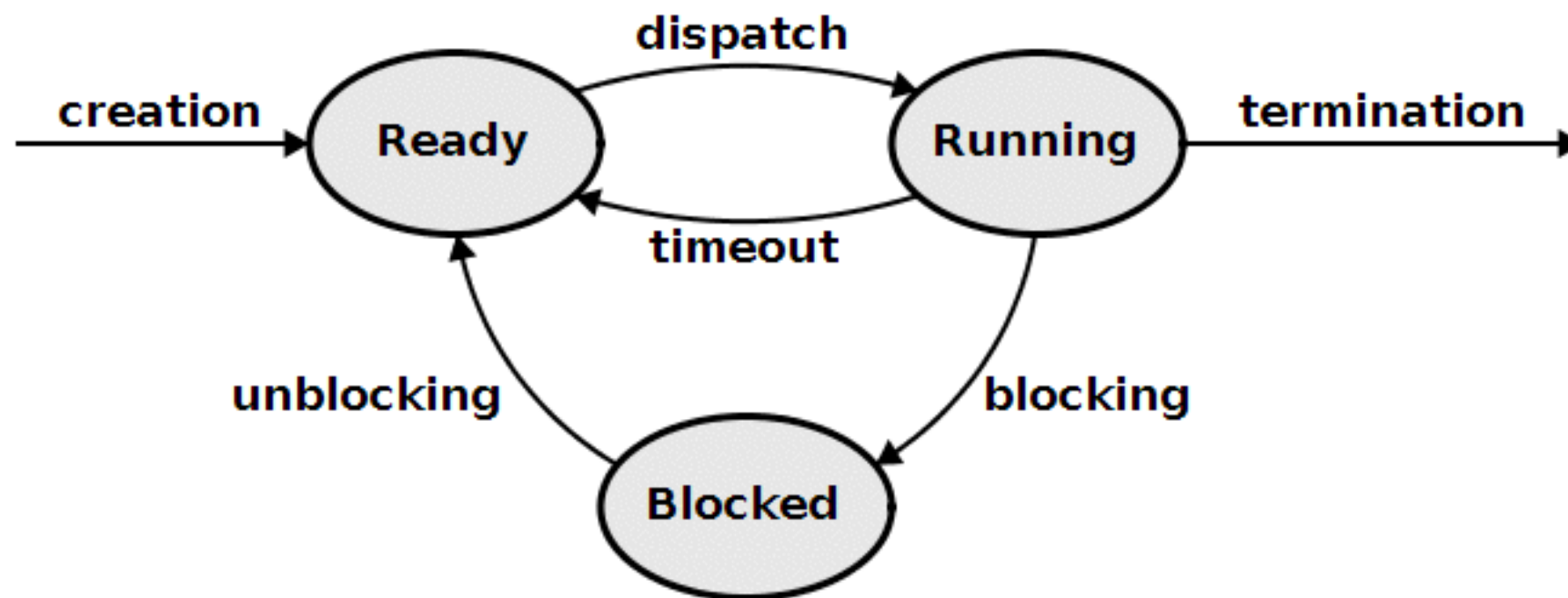
O que vimos na aula passada

- **Gerenciamento de Processos**
- **Introdução ao Docker**

Objetivo da Aula

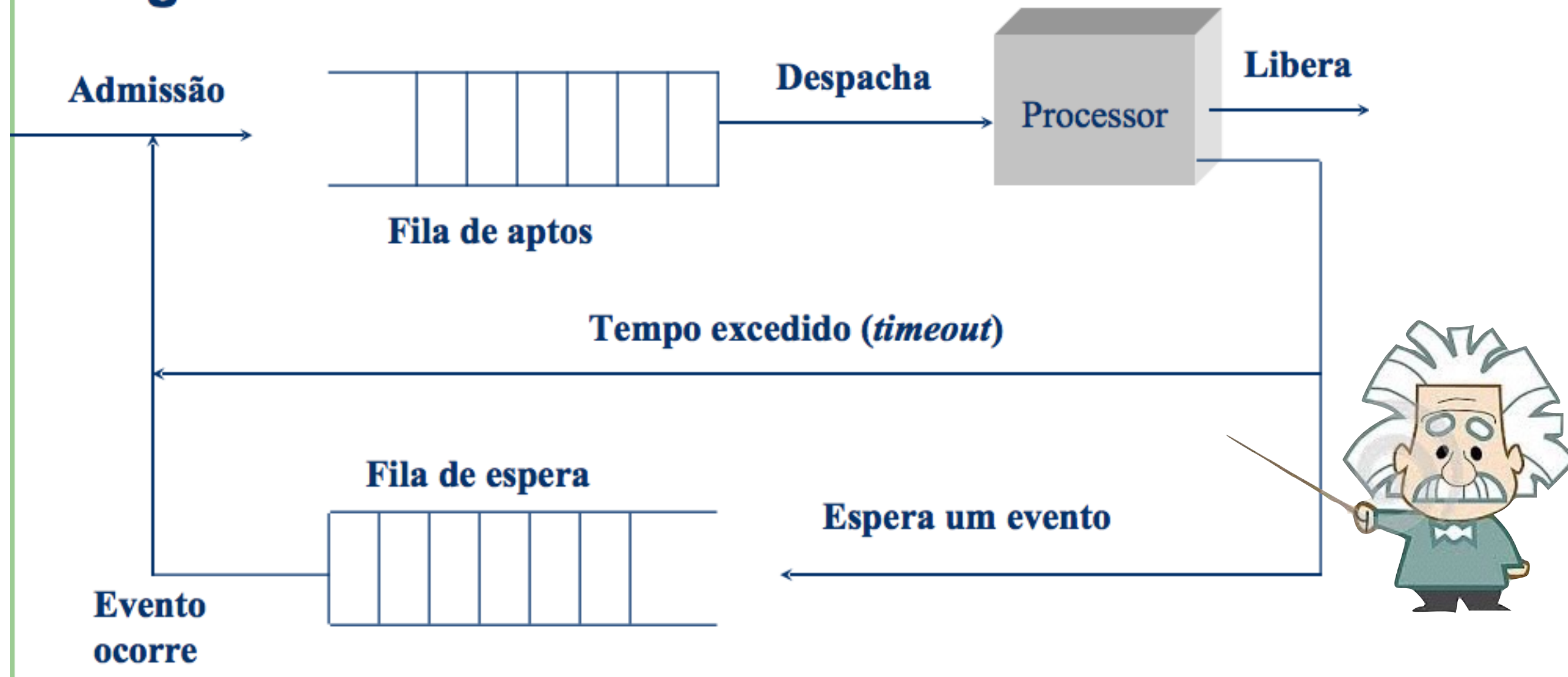
- **Gerenciamento de Processos**
- **Gerenciamento de Processos no Docker**

Ciclo de Vida de Processos



Fila de atendimento de processos

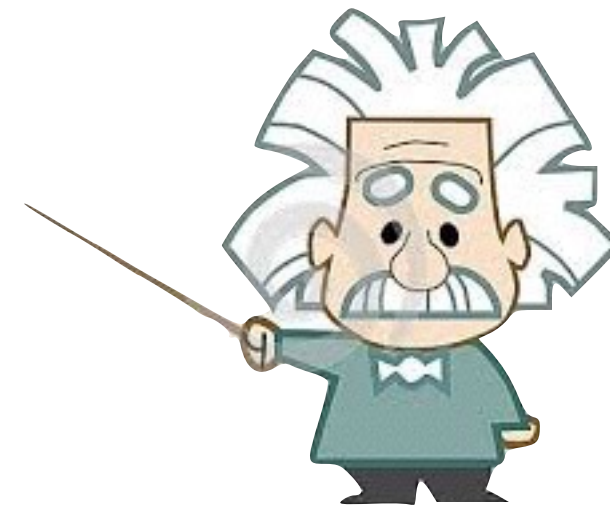
Diagrama de fila



Escalonamento de Processos

Escalonamento

- Determinação da ordem em que processos alocam o processador
- SO usa algum critério para escolher um processo apto para rodar
- SO pode também tirar o processador de um processo que estiver rodando



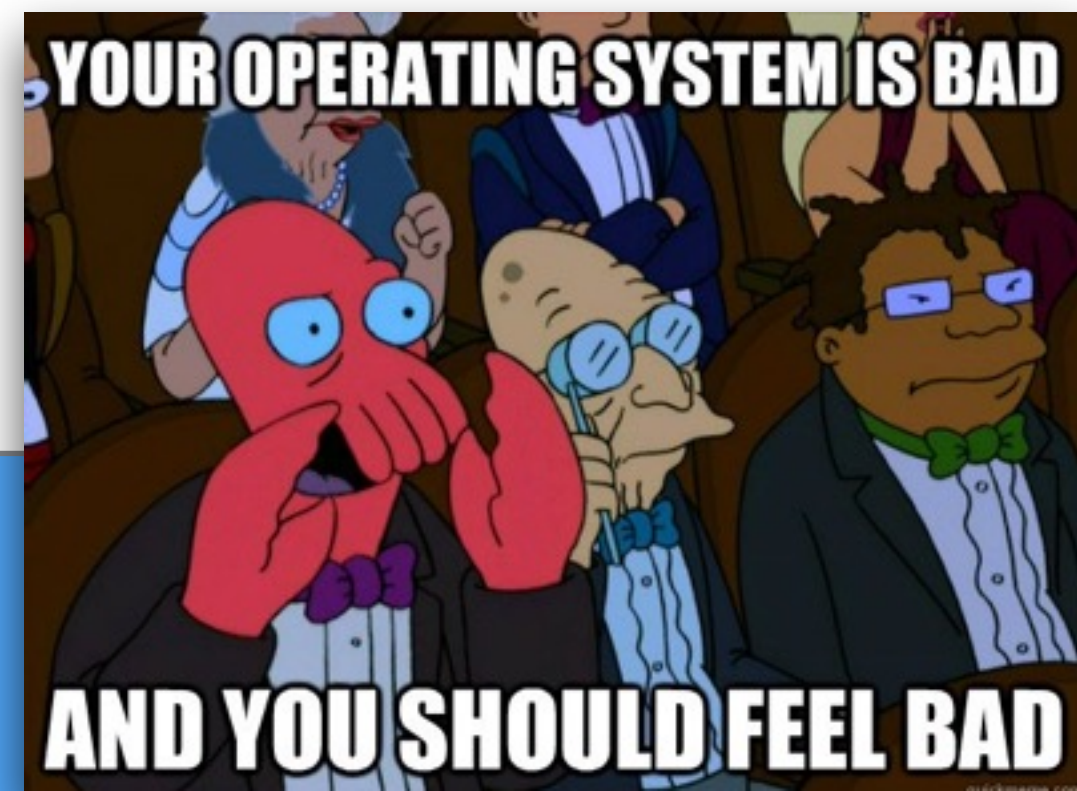
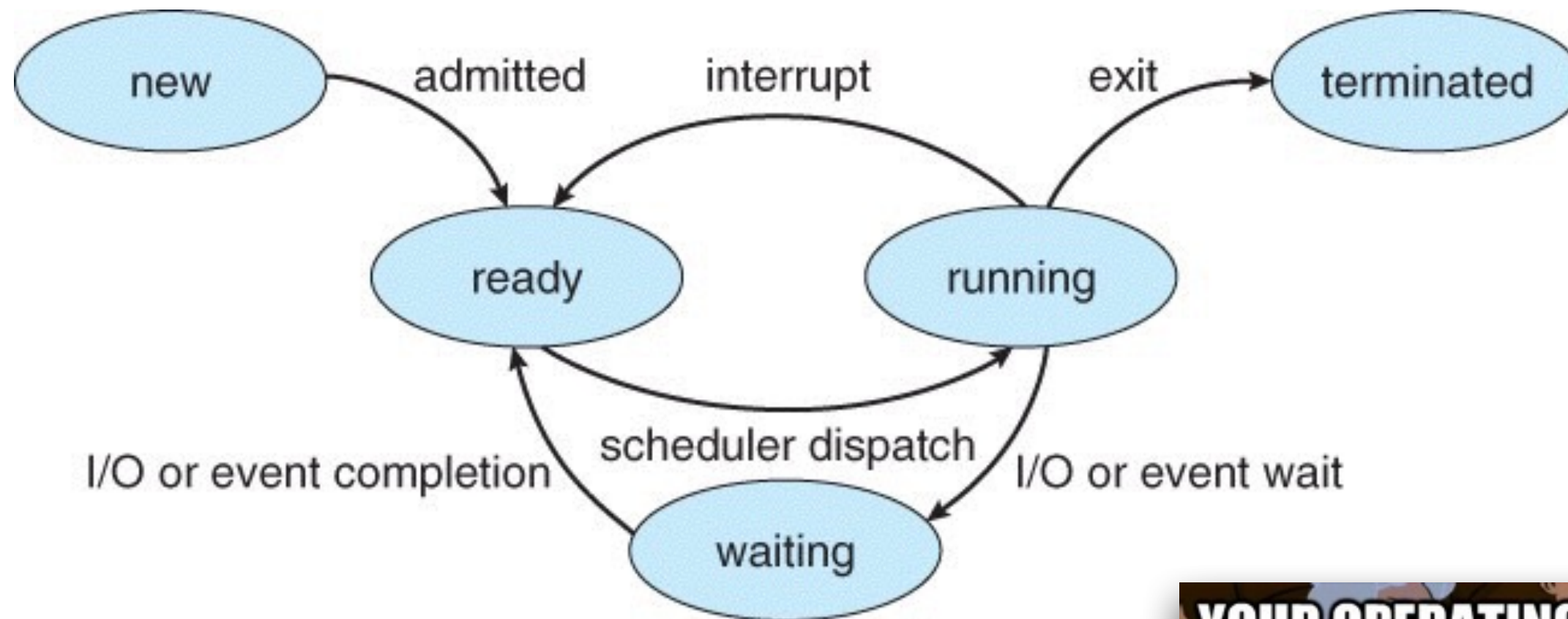
Escalonamento de Processos

Escalonamento

- ***Escalonamento não-preemptivo***
 - Processo só perde o processador se terminar ou entrar em estado de espera
- ***Escalonamento preemptivo***
 - Sistema operacional pode retomar o processador mesmo contra a vontade do processo

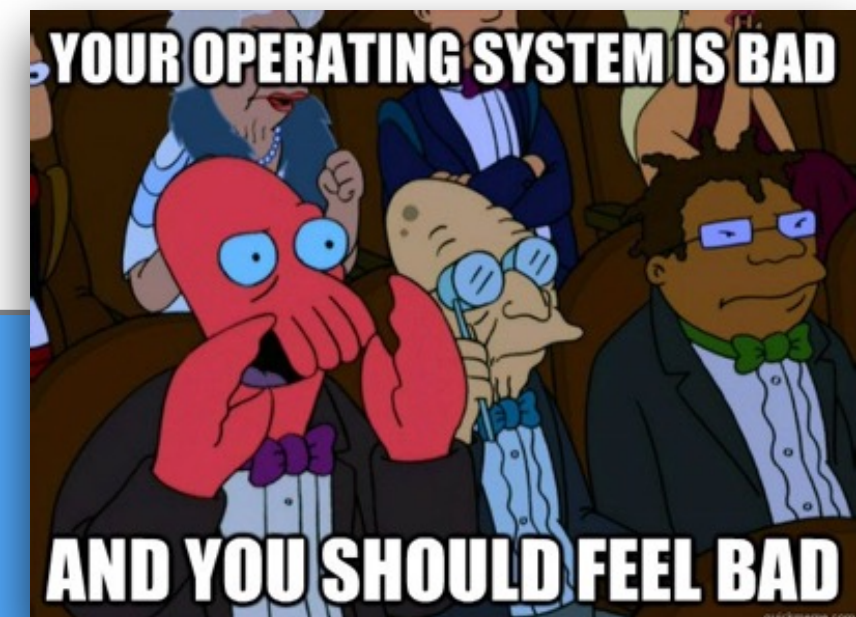


Escalonamento de Processos



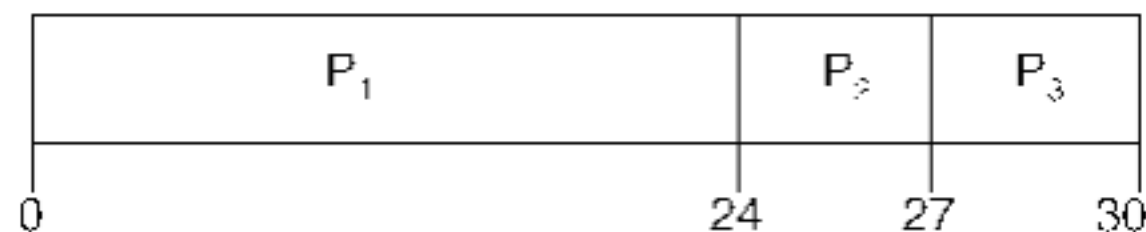
Escalonamento de Processos

- **Critérios para um bom escalonamento**
 - **Eficiência:** ocupação do processador
 - **Justiça:** todos os processos têm chance de executar em um ciclo de escalonamento
 - **Tempo de resposta:** tempo entre a ocorrência de um evento e a reativação do processo que o aguarda (*IMPORTANTE !*)
 - **Throughput:** número de processos concluídos por unidade de tempo

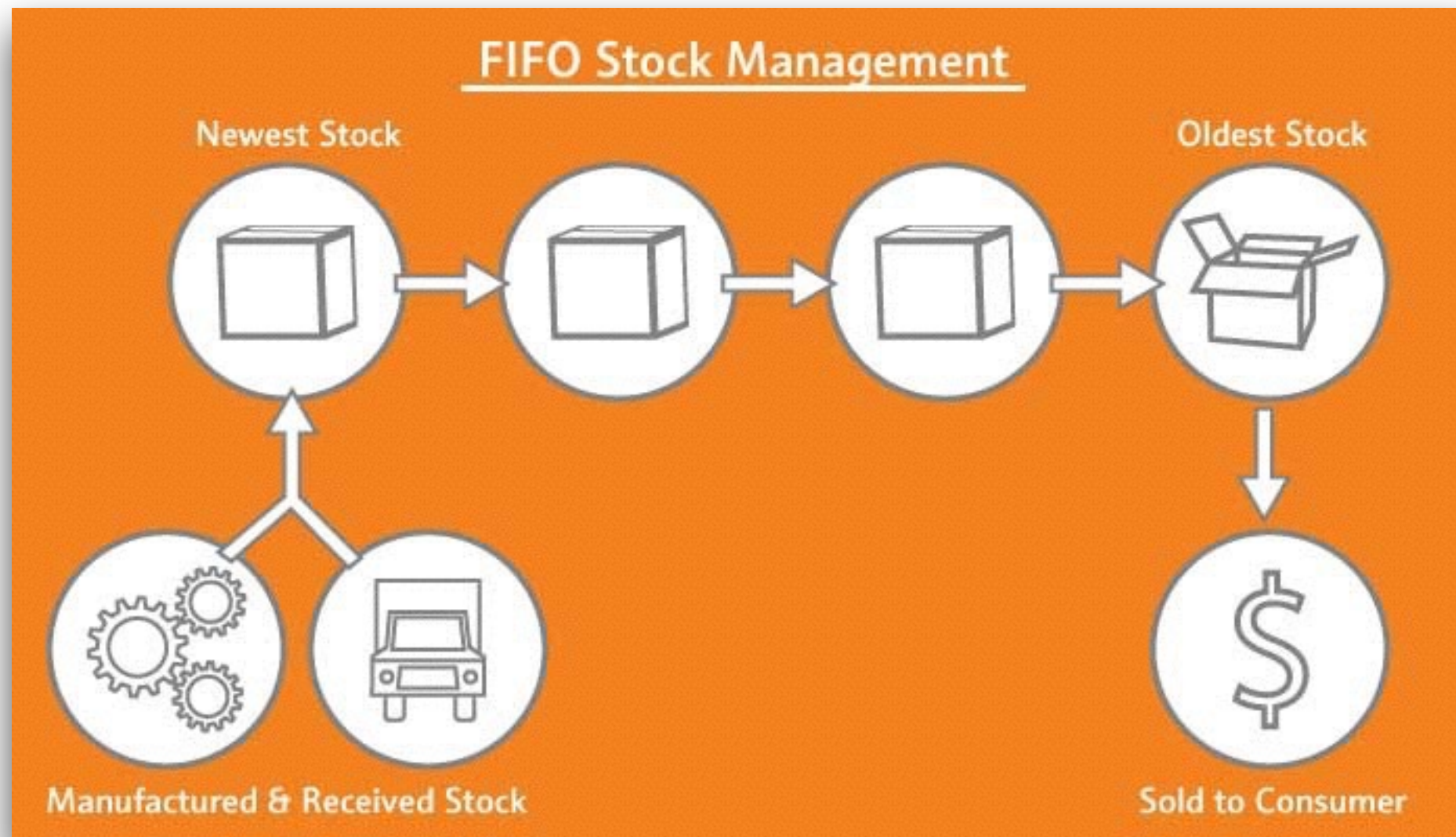


Escalonamento de Processos

- **FIFO** (*First In First Out*) ou **FCFS** (*First Come First Served*)
 - Primeiro que entra é o primeiro que sai da fila
 - Ex: Processos P1, P2 e P3 têm tempo de CPU de 24, 3 e 3 unidades de tempo respectivamente, e chegam nesta ordem.



Escalonamento de Processos



Escalonamento de Processos



Escalonamento de Processos

• **FIFO (cont.)**

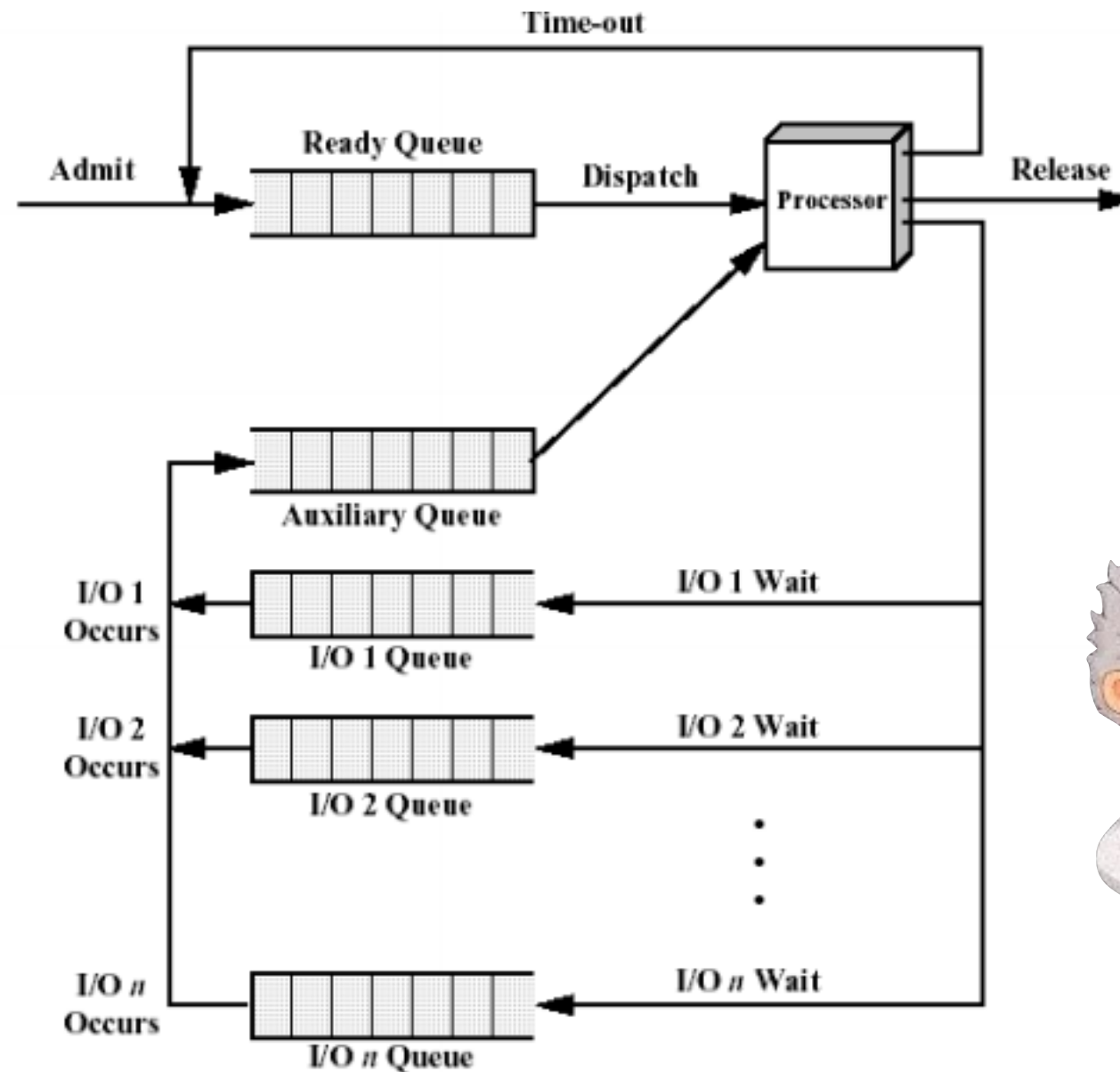
- Tempo de execução médio: $T = \frac{1}{N} \sum_{i=1}^N (N - i + 1) \cdot T_i$
- Minimização desse tempo: $T_i < T_{i+1}$
- *FIFO é muito simples, mas não atende os requisitos de bom escalonamento (por que ?)*



Escalonamento de Processos

- **RR (*Round Robin*)**

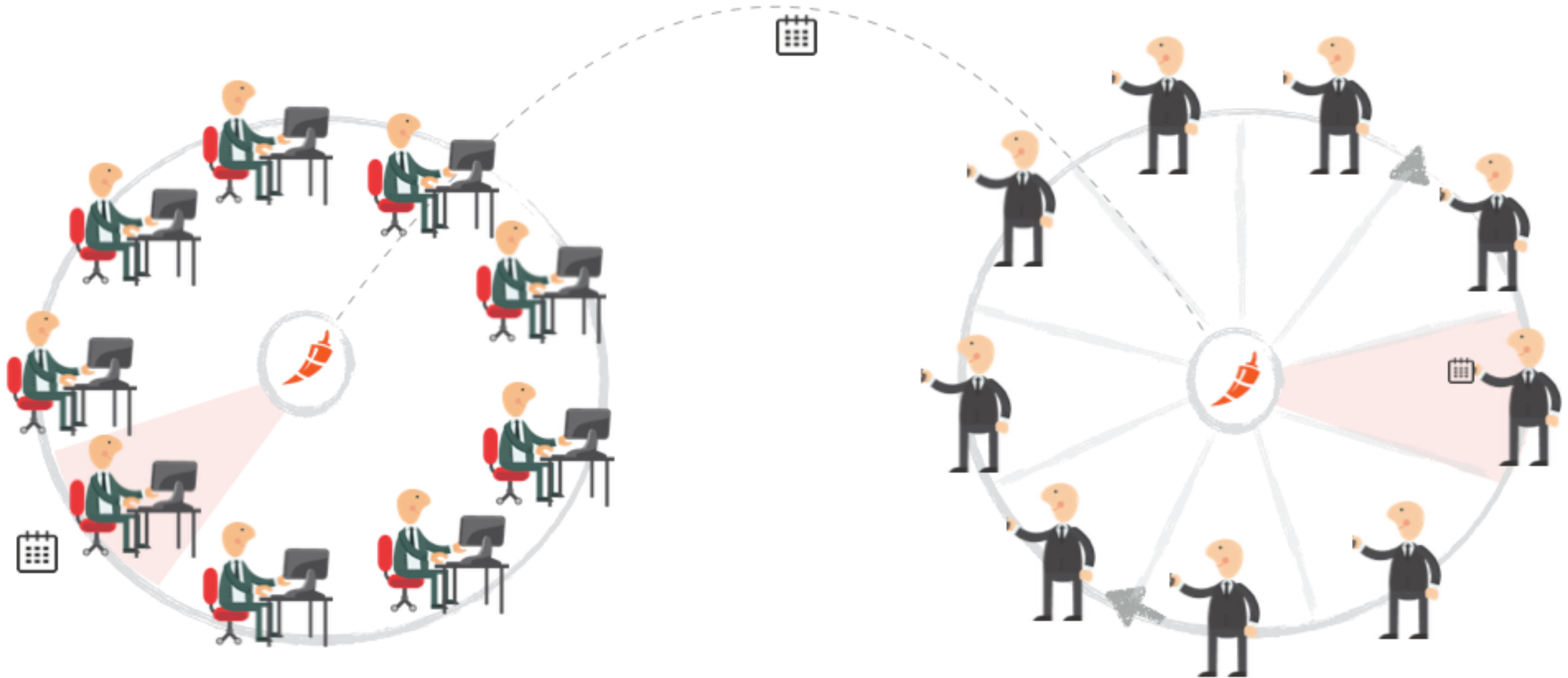
- Cada processo usa a CPU por até 1 quantum de tempo por ciclo
- Fila de aptos segue FIFO



Escalonamento de Processos



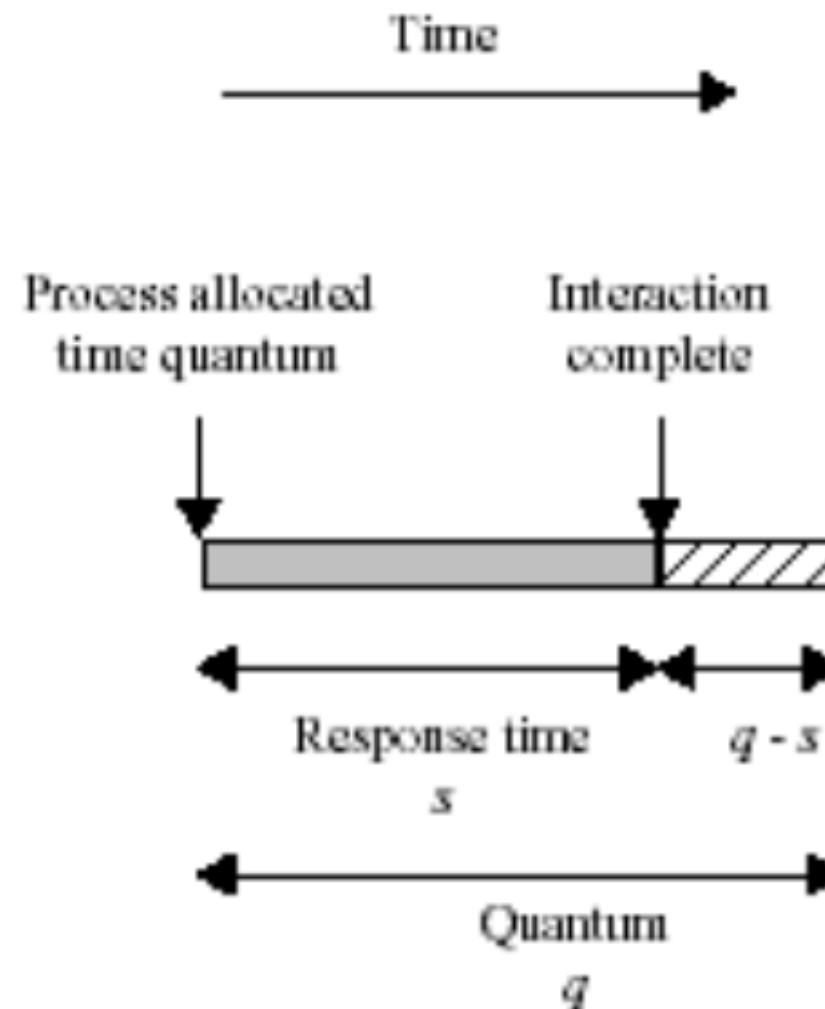
Escalonamento de Processos



Escalonamento de Processos

• **RR (cont.)**

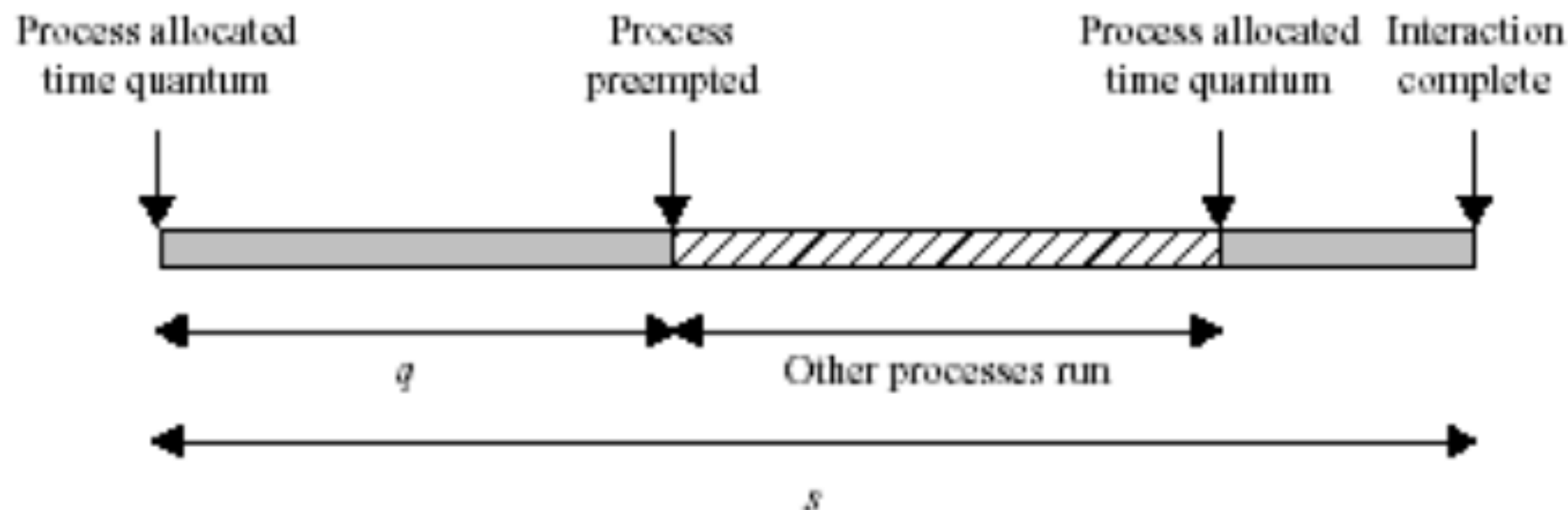
- *quantum* deve ser maior ou igual ao tempo de interação médio (tempo de CPU médio)



Escalonamento de Processos

- **RR (cont.)**

- Se quantum for muito curto, aumenta-se o tempo de execução total:



Escalonamento de Processos

RR (cont.)

- Casos limites:
 - $q \rightarrow 0$: Cada um dos N processos enxerga um processador com $1/N$ de sua capacidade
 - Na verdade, *overhead* do chaveamento de contexto domina o processamento
 - $q \rightarrow \infty$: Escalonamento degenera para FIFO



Escalonamento de Processos

Com prioridades

- Cada processo possui uma prioridade (um número inteiro)
- A fila de aptos é ordenada de acordo com as prioridades



Escalonamento de Processos

- **Com prioridades (cont.)**
 - Considerado o mais genérico
 - Qualquer outro algoritmo pode ser implementado usando prioridades
 - RR e FIFO : todos processos com igual prioridade
 - Prioridades podem variar com o tempo
 - Podem se adaptar ao comportamento dos processos



Escalonamento de Processos

- **Com prioridades (cont.)**
 - ***Prioridades dinâmicas:*** o que aconteceria se o histórico de uso do processador fosse usado como prioridade ?
 - Ex: ***prioridade = tempo de uso / tempo total***
 - Que tipo de processo seria priorizado com prioridades assim ?
 - Que critérios de bom escalonamento seriam respeitados com isto ? E quais não seriam ?



Escalonamento de Processos

- **Com prioridades (cont.)**
 - Muitos SO usam prioridades dinâmicas, dando maior prioridade para processos I/O bound
 - Ex: Linux, Windows Xp/Vista, Sun Solaris, MacOS X, ...
 - Com isto, priorizam processos **interativos**
 - Porém, o que dizer de processos que processam *streams multimedia* ?
 - Ex: MP3 player, Video player, ...



Escalonamento de Processos

- **Descrição de processos**

- Processos possuem um ID único (PID)
- Possuem um usuário dono (UID) e um usuário efetivo (EUID), assim como grupo dono (GID) e grupo efetivo (EGID)
- Prioridade base (*nice*) pode ser alterada
- Toda a memória e objetos de sua posse estão protegidos de outros processos



Escalonamento de Processos

Parent

```
main()    pid = 3456
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void ChildProcess()
{
    .....
}

void ParentProcess()
{
    .....
}
```

Child

```
main()    pid = 0
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void ChildProcess()
{
    .....
}

void ParentProcess()
{
    .....
}
```



SHELL - Processo no Linux

D (uninterruptible sleep) - Process is sleeping and cannot be bring back until an event such as I/O occurred. For example, process foo is a process waiting for keyboard interrupt.

R (running) - Process is running or executing.

S (sleeping) - Process is not running and is waiting for an event or a signal.

T (traced or stopped) - Process is stopped by signals such as SIGINT or SIGSTOP.

Z (zombie or defunct) - Processes marked <defunct> are dead processes (so-called "zombies") that remain because their parent has not destroyed them properly. These processes will be destroyed by init if the parent process exits.

SHELL - Processo no Linux

ps -C processName -o pid=,cmd,stat

ps -C firefox-bin -o pid=,cmd,stat

ps -C lighttpd -o pid=,cmd,stat

ps -C php-cgi -o pid=,cmd,stat

```
      CMD                                STAT
7633 /opt/firefox/firefox-bin           Sl

      CMD                                STAT
32082 /usr/sbin/lighttpd -f /etc/       S
32326 /usr/sbin/lighttpd -f /etc/       S

      CMD                                STAT
1644 /usr/bin/php-cgi                   S
31331 /usr/bin/php-cgi                   S
31332 /usr/bin/php-cgi                   S
31538 /usr/bin/php-cgi                   S
```


SHELL - Visualizando processos

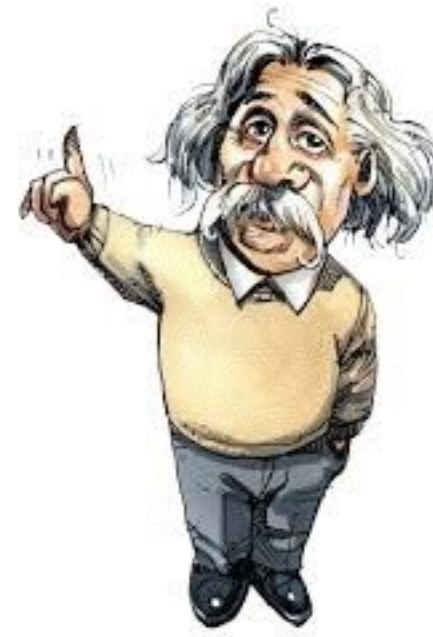
ps

ps aux | less

ps aux | grep "process-name"

ps aux | grep "httpd"

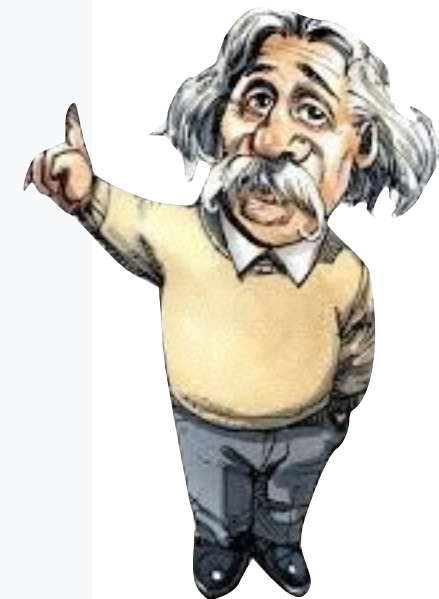
ps alx | grep "mysqld"



SHELL - Visualizando processos

pstree

```
init└─acpid
    ├─apache2──6*[apache2]
    ├─atd
    ├─atop
    ├─avahi-daemon──avahi-daemon
    ├─bonobo-activati──{bonobo-activati}
    ├─console-kit-dae──63*[{console-kit-dae}]
    ├─cron
    ├─2*[dbus-daemon]
    ├─dbus-launch
    ├─dd
    ├─deluge──5*[{deluge}]
    ├─dhclient
    ├─dnsmasq
    ├─evince──{evince}
    ├─firefox──run-mozilla.sh──firefox-bin──27*[{firefox-bin}]
    ├─gconfd-2
    ├─gdm──gdm└─Xorg
    │           └─gnome-session└─gnome-panel
    │                               └─gpg-agent
    │                                   └─metacity
```



SHELL - Visualizando processos

pgrep

pgrep -u vivek php-cgi



SHELL - Finalizando processos

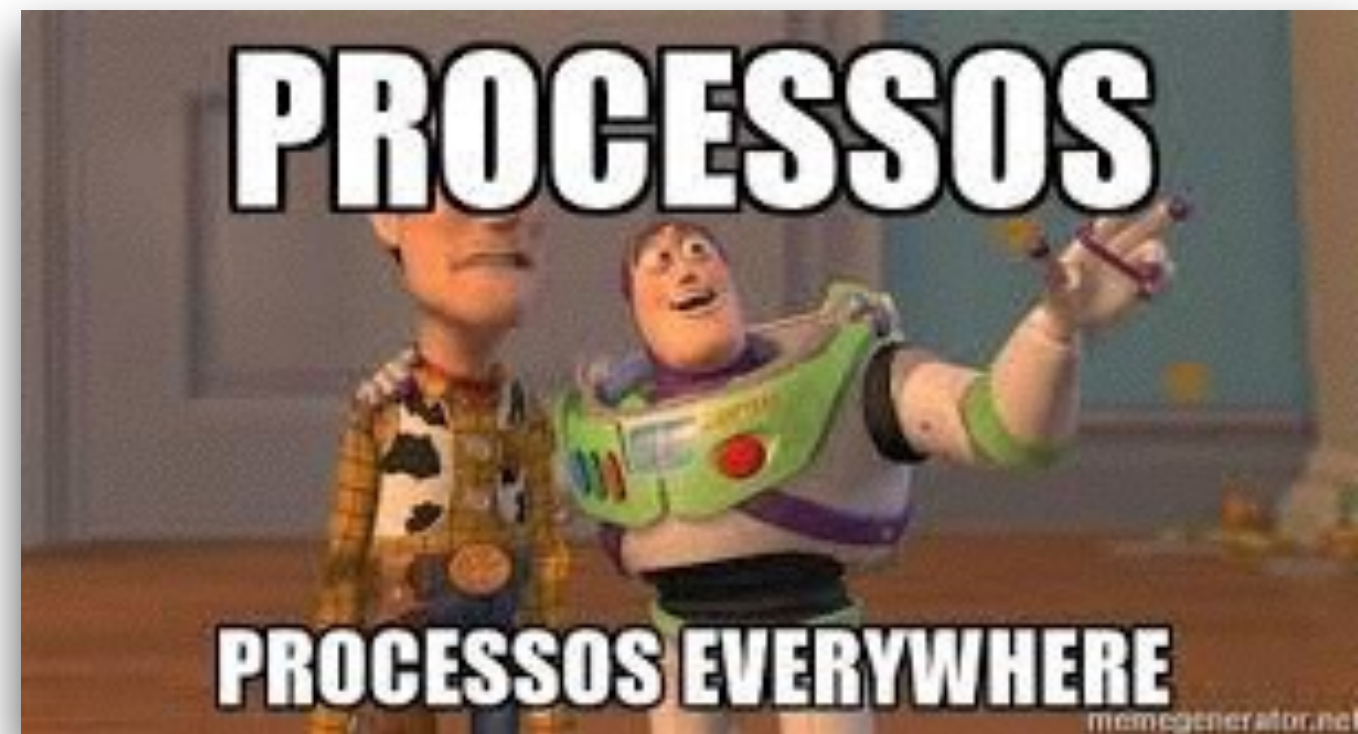
```
kill -9 1234
```

OR

```
kill -KILL 1234
```

OR

```
kill -SIGKILL 1234
```



SHELL - Matando processos pelo Nome

killall - kill processes by name

killall sends a signal to all processes running any of the names (and parent), enter:

```
killall processName  
killall firefox-bin
```

To send a KILL signal to firefox, enter:

```
killall -s SIGKILL firefox-bin
```



SHELL - Matando processos pelo Nome

```
obelix[3] > jobs
```

```
[1] + Suspended
```

```
[2] + Suspended
```

```
make_noise
```

```
vi readme
```


Ciclo de vida de um processo

Bóson Treinamentos 2013

Controle de Tarefas

Exemplo:
gimp &

Para colocá-lo em background, digite Ctrl+Z (sinal TSTP, 20), e em seguida digite o comando **bg** para reiniciar a tarefa em background.

Para colocar a tarefa novamente em foreground, digite o comando **fg**.

Exemplo:

vi /home/fabio/arq1

Ctrl+Z

bg

#Agora a tarefa está em background. Visualize com ps, e use o terminal para #digitar outros comandos. Para voltar a tarefa ao foreground então:

fg

Conhecimentos Adquiridos

- Aprender conceitos de gerenciamento de Processos
- Aprender conceitos iniciais de shell script

