

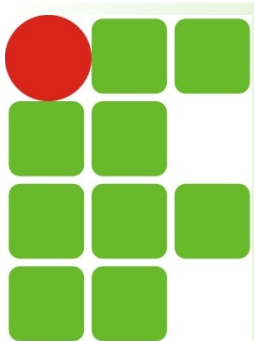
**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
RIO GRANDE DO NORTE  
Campus Currais Novos



# Introdução a Sistemas Abertos

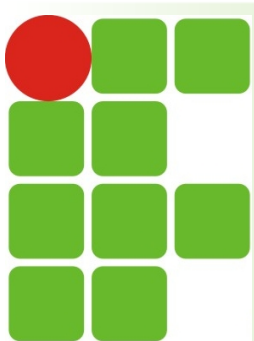
## Aula 07 – Comandos Linux Gerenciamento de Processos

Prof. Diego Pereira <diego.pereira@ifrn.edu.br>



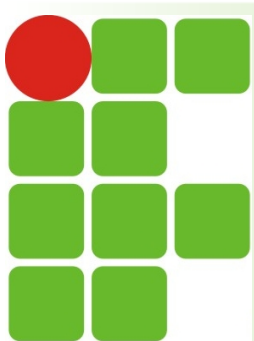
# Objetivos

- Entender os principais conceitos envolvidos com processos em sistemas operacionais;
- Aprender os estados de um processo;
- Conhecer alguns comandos para auxiliar no gerenciamento de processos;



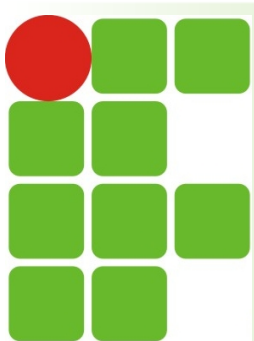
# Introdução

- Todo sistema operacional fornece uma interface para que os usuários tenham algum controle sobre os programas que estão sendo executados;
- No Linux isto ocorre através do gerenciamento de processos via prompt de comando;



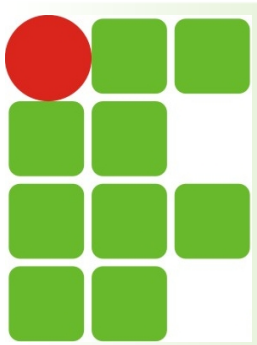
# Processo

- Definição
  - Processo é um programa em execução;
  - Do ponto de vista do S.O. é a estrutura responsável pela manutenção de todas as informações necessárias para a execução de um programa;



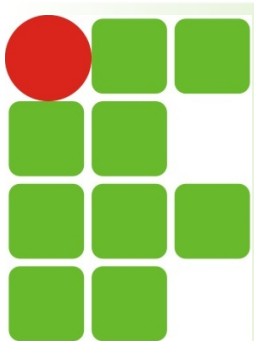
# Processo

- Processos são entidades independentes;
- Cada um possui permissões de acesso e atributos(características);
- Ex: PID(Process IDentification)
- O S.O. é responsável em gerenciar os processos de forma a otimizar a utilização da CPU;
- Caso o processo necessite ler/escrever dados no disco rígido, ele fica em estado *waiting* até que a operação seja completada;



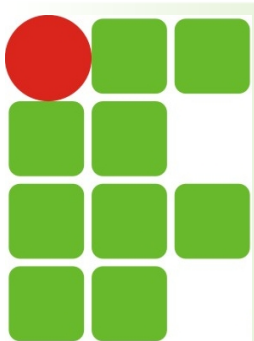
# Processo

- No diretório /proc é criado um subdiretório para cada processo em execução;
- Os nomes dos subdiretórios são seus PIDs;
- Dentro desse subdiretório existem diversos arquivos com informações sobre o processo;



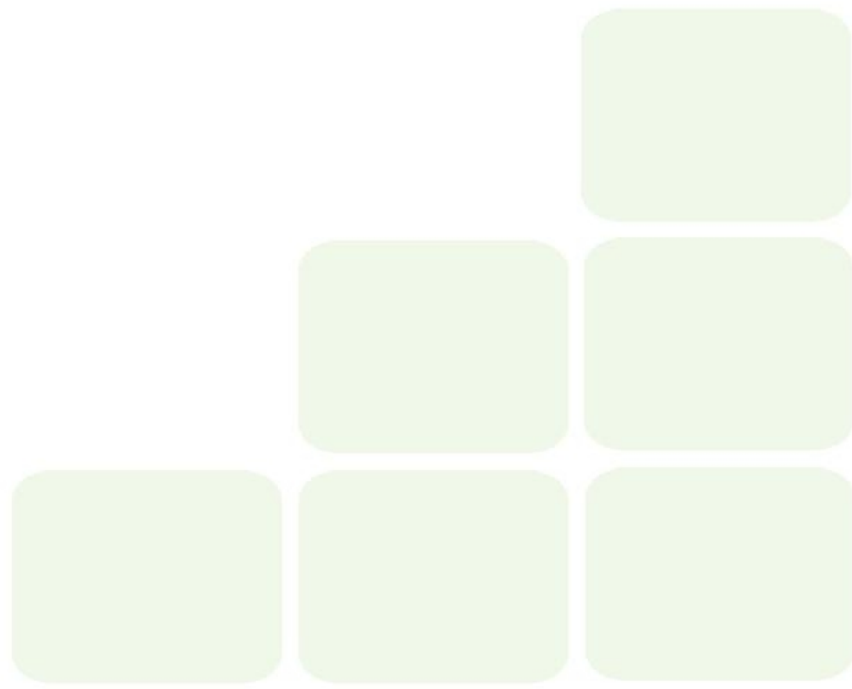
# Alguns Arquivos

Arquivo	Descrição
cmdline	guarda a linha de comando usada para iniciar o processo
environ	mostra as variáveis de ambiente utilizadas pelo processo
exe	é um link (apontador) para o executável daquele processo
fd	diretório contendo todos os descritores de arquivo
limits	as restrições impostas ao processo (/etc/security/limits.conf)
maps	mapa de memória para os executáveis e bibliotecas
root	apontador para o diretório raiz do processo
mounts	informação sobre os sistemas de arquivo
mem	memória utilizada pelo processo
sched	informações sobre o escalonador e o escalonamento
stat	status do processo
statm	status da memória do processo

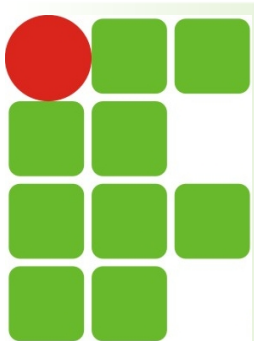


# Alguns Comandos

- Os comandos via shell também são uma maneira eficaz de obter informações sobre os processos;
- Exemplos:
  - ps;
  - pstree;
  - top;

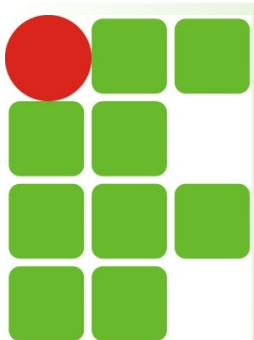






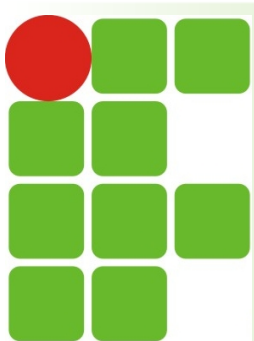
# Prioridade

- Também é possível alterar a prioridade de execução dos processos;
- Varia entre -20(maior prioridade) e 19(menor prioridade);
- Apenas o root pode atribuir prioridades negativas a processos;
- Comandos utilizados são o nice e o renice;



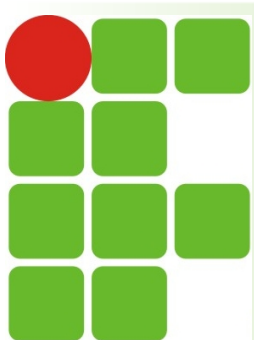
# Atributos de um processo

- Um processo possui diversos atributos que controlam sua execução;
  - PID
    - Process IDentification(Identificação do processo), identifica um processo em execução, não pode ser repetido, é exclusivo.
  - PPID
    - Parent Process IDentification(Identificação do processo-pai), identifica o processo-pai que gerou o processo-filho;



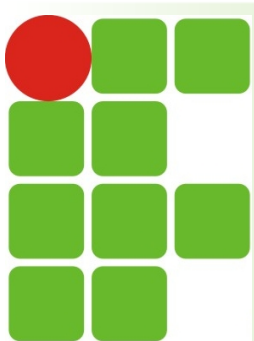
# Atributos de um processo

- **UID**
- **User IDentification**(Identificação do Usuário), Identifica o usuário que criou o processo;
- **GID**
- **Group IDentification**(Identificação do Grupo), identifica o grupo ao qual pertence o processo;



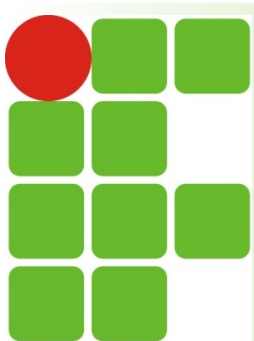
# Atributos de um processo

- EUID
- Effective User IDentification (Identificação do Usuário Efetivo), quando um programa é executado com seu setuid ligado (`chmod +s`), o EUID passa a ser do dono do arquivo executável, em vez do UID do usuário que está executando o programa;
- EGID
- Effective Group IDentification (Identificação do Grupo Efetivo);



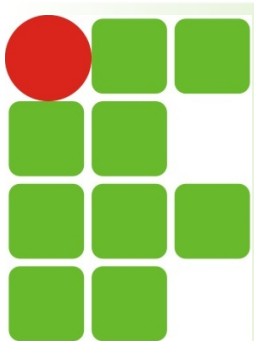
# Sequência de Execução de um Processo

- Quando um processo está sendo executado, ele passa por diversos estados, são eles;
  - Ready(pronto)
  - Logo que é iniciado é colocado no estado ready, ou seja, está pronto para entrar em execução mas irá aguardar o scheduler(escalaonador de tarefas) do kernel decidir quando ele irá entrar em execução;



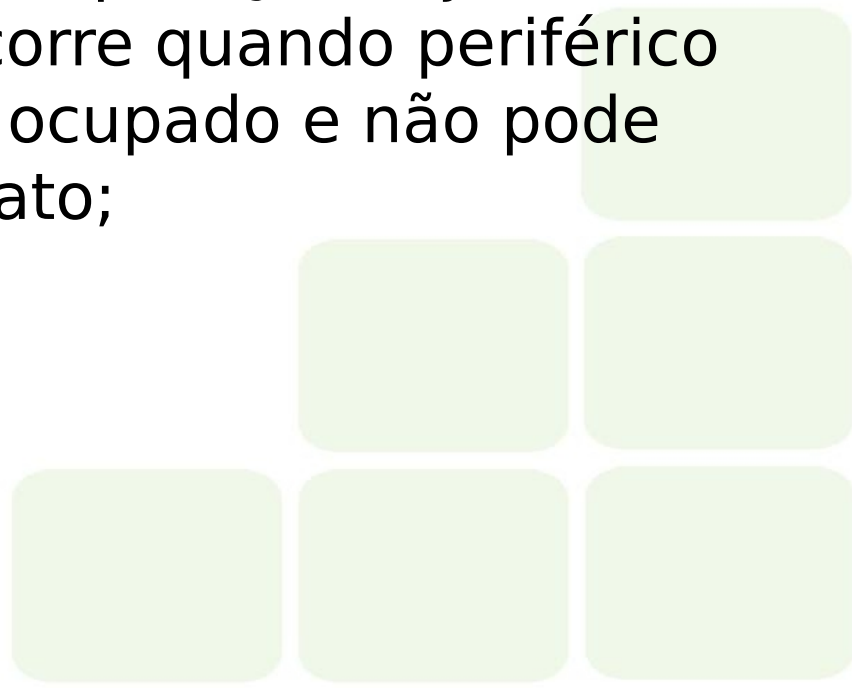
# Sequência de Execução de um Processo

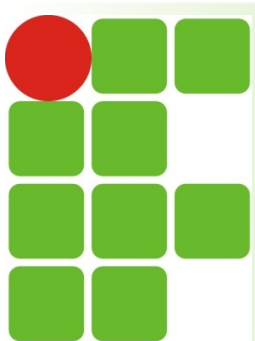
- Running(execução)
- O kernel decide que o processo irá entrar em execução, tal decisão é tomada de acordo com a prioridade e a fila de execução;
- Quando seu time slice(fatia de tempo) esgotar, ele volta para o estado de ready e aguarda um novo escalonamento;



# Sequência de Execução de um Processo

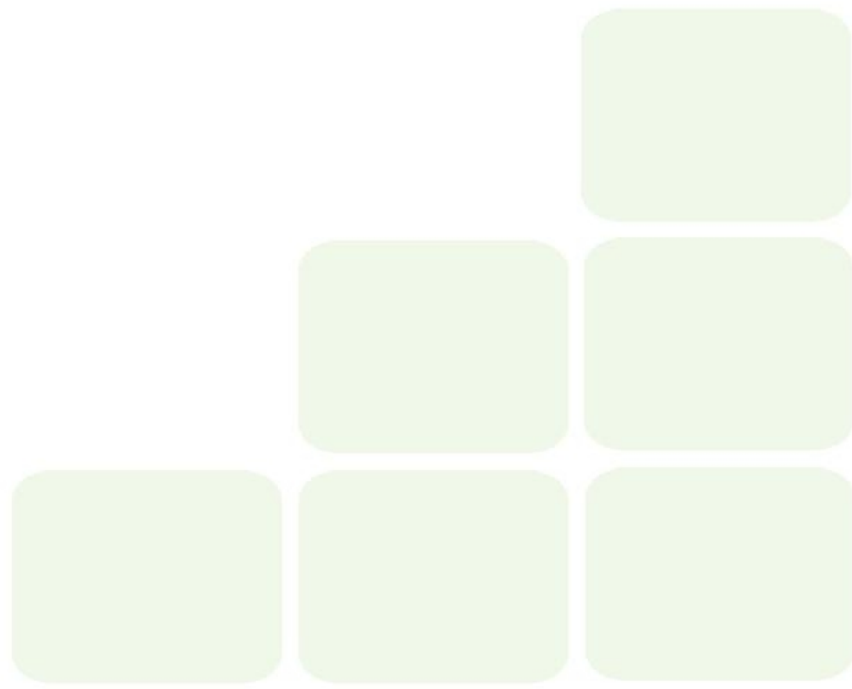
- Waiting(espera)
- Caso o processo necessite de uma operação de entrada/saída ele pode ser colocado em waiting até que essa operação seja completada, isso ocorre quando periférico em questão estiver ocupado e não pode responder de imediato;





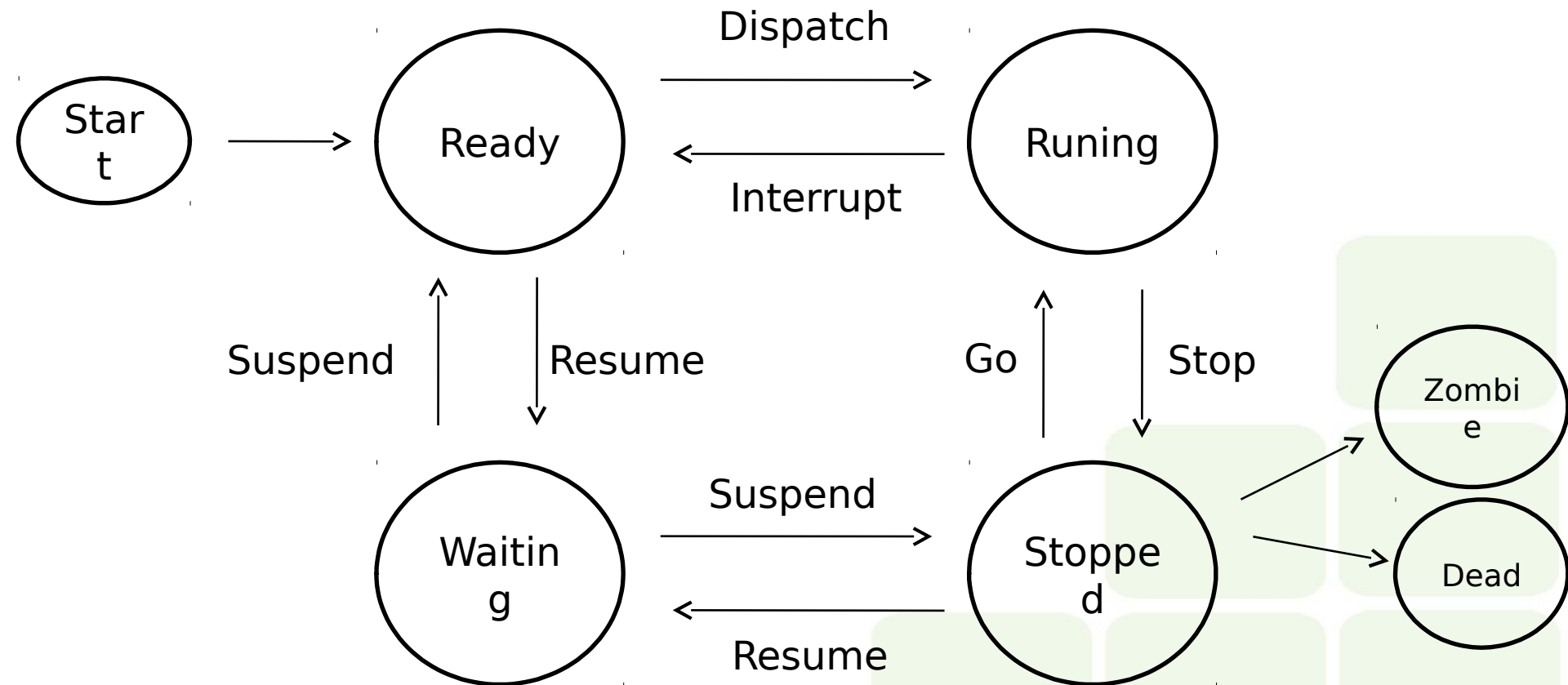
# Sequência de Execução de um Processo

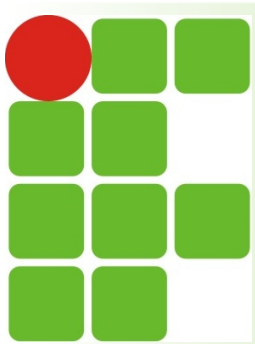
- Dead(Morto - Finalizado) e Zombie
- No final da execução, o processo pode ser morto(comando kill) ou zombie(caso seu parent - processo pai - tenha sido morto);





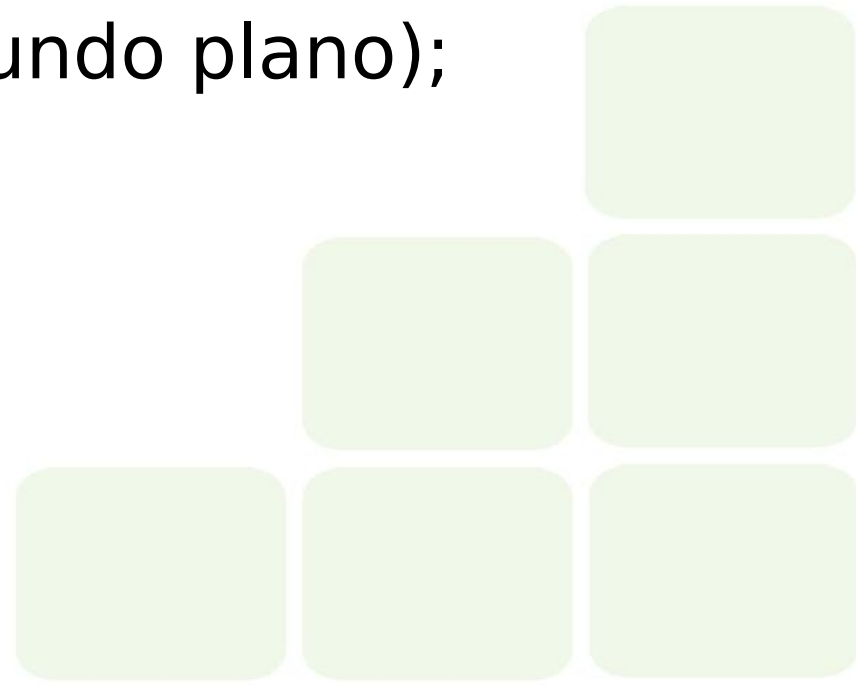
# Sequência de Execução de um Processo

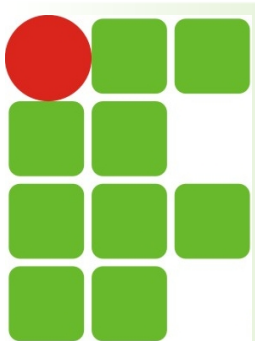




# Classificação de processos

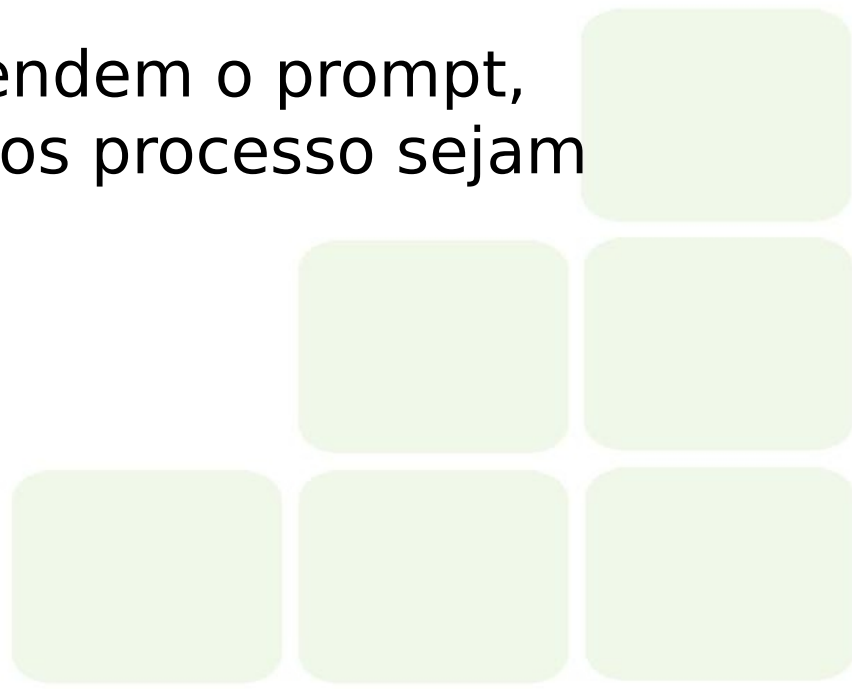
- Quanto à execução, os processos podem ser classificados em:
  - Foreground(primeiro plano);
  - Background(segundo plano);

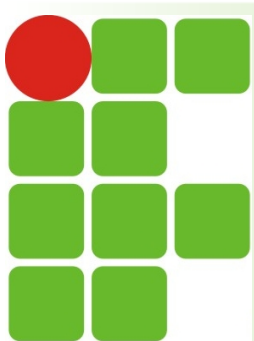




# Classificação de processos

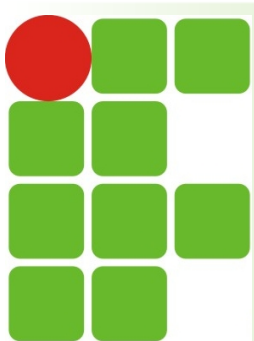
- Foreground(primeiro plano)
- São inicializados no terminal de comandos, podem interagir com os usuários e exibem sua execução no monitor de vídeo;
- Esses processos prendem o prompt, impedindo que outros processo sejam inicializados;





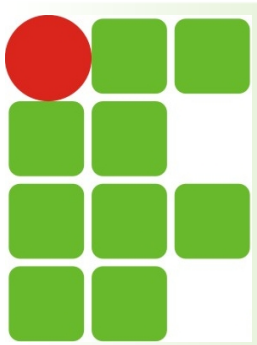
# Classificação de processos

- Background(segundo plano)
- Também são inicializados no terminal de comandos, mas não interagem com usuários e não exibem sua execução no monitor de vídeo;
- Esse processos não prendem o prompt e permitem que outros processos sejam inicializados pelo terminal;



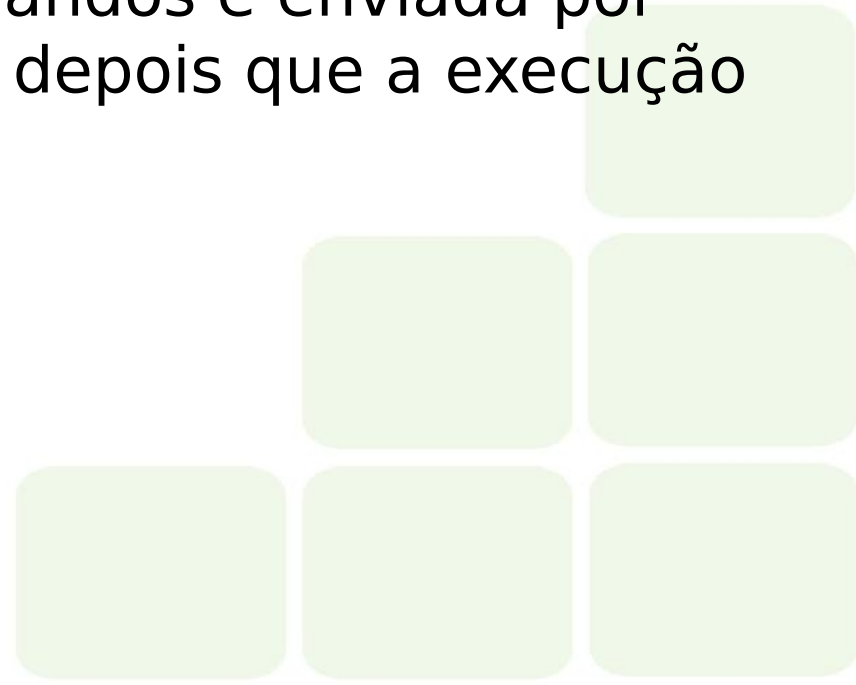
# Classificação de processos

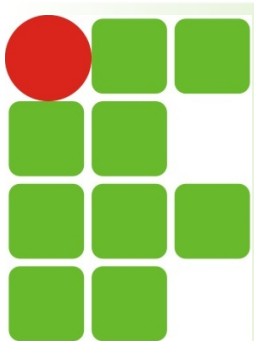
- Quanto ao tipo, os processos podem ser classificados em:
  - Processos Interativos
  - São iniciados a partir de uma sessão de usuário no terminal de comandos e controlados por ele;
  - Quando um comando no shell é executado, um processo é executado em foreground(primeiro plano);



# Classificação de processos

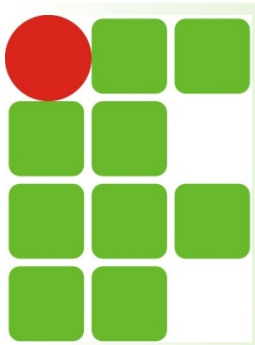
- Processos em lote(batch)
- São controlados pelos comandos at, batch e cron;
- A saída desses comandos é enviada por mail para o usuário depois que a execução for concluída;





# Classificação de processos

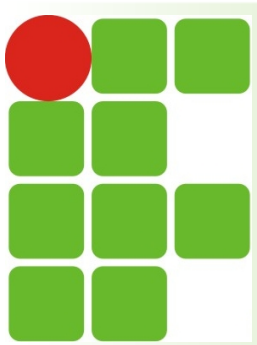
- Daemons
- Processos servidores normalmente executados quando o Linux é inicializado, permanecendo em execução enquanto o sistema estiver em funcionamento;
- Aguardam em background que algum outro processo solicite seu serviço;
  - Ex: apache2, postfix, bin9...



# Controle de tarefas (job control)

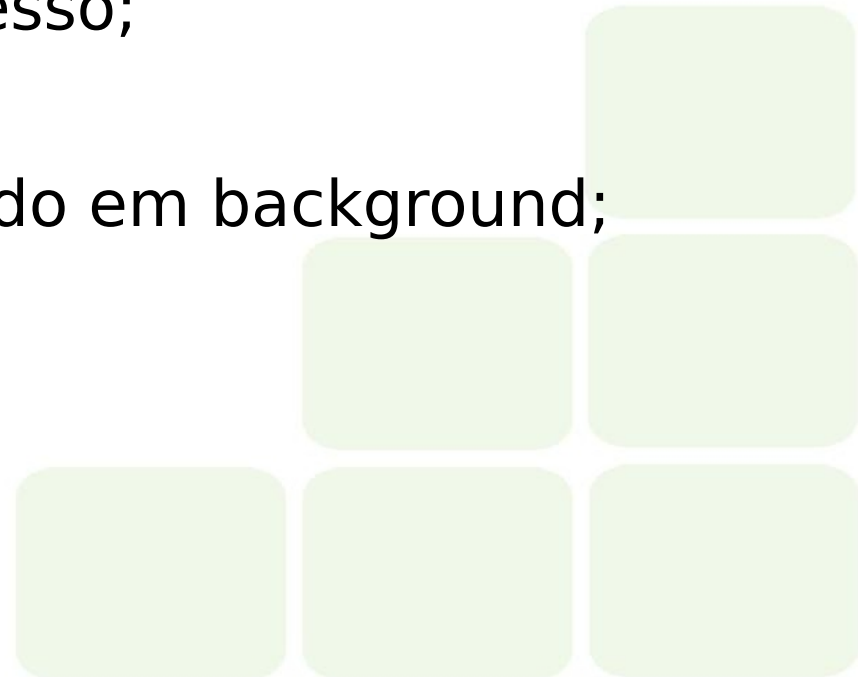
- Refere-se à habilidade de suspender a execução de processos e retomar a sua execução posteriormente;
  - O Shell associa um ou mais processos a cada job inicializado por ele

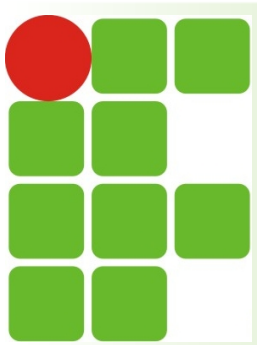




# Comandos para Gerenciamento de Processos

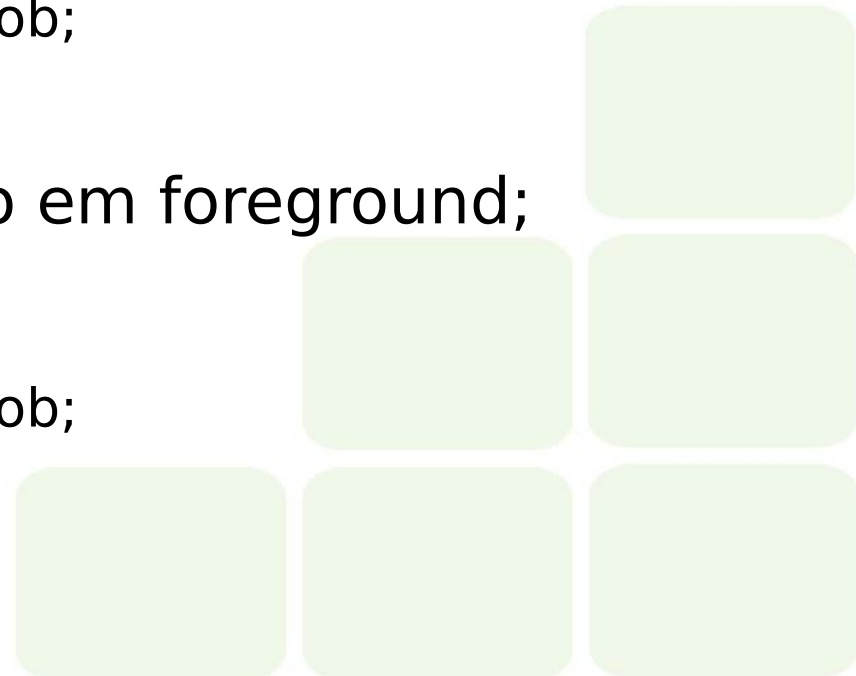
- CTRL+C
  - Aborta um processo;
- CTRL+Z
  - Suspende um processo;
- &
  - Executa um comando em background;
- comando &
  - vi &

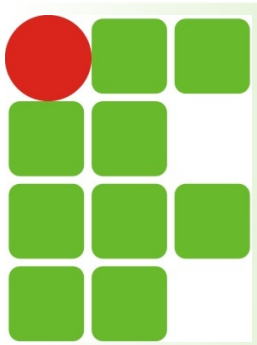




# Comandos para Gerenciamento de Processos

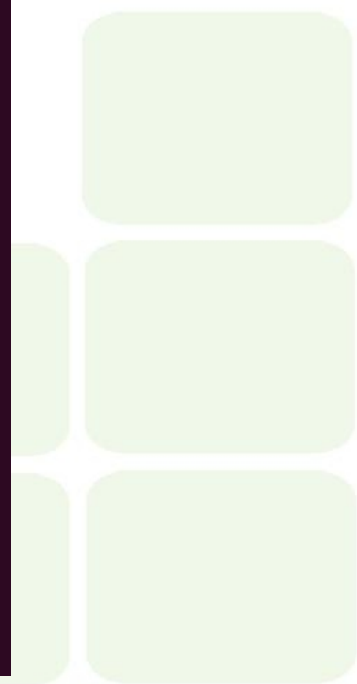
- `bg`
- Coloca um processo em background;
- `bg %id`
  - `%id` é o número do job;
- `fg`
- Coloca um processo em foreground;
- `fg %id`
  - `%id` é o número do job;

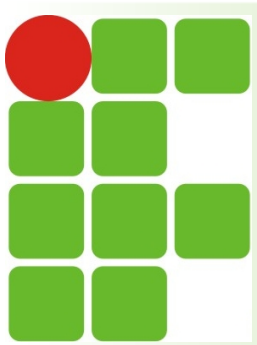




# Comandos para Gerenciamento de Processos

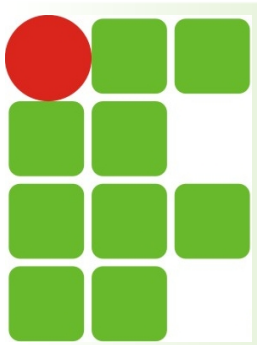
```
root@diego-VirtualBox: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
root@diego-VirtualBox:~# vi  
[1]+  Parado          vi  
root@diego-VirtualBox:~# bg %1  
[1]+ vi &  
[1]+  Parado          vi  
root@diego-VirtualBox:~# jobs  
[1]+  Parado          vi  
root@diego-VirtualBox:~# fg %1  
vi  
root@diego-VirtualBox:~# jobs  
root@diego-VirtualBox:~#
```





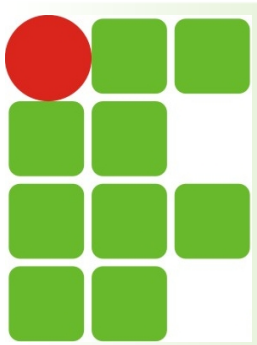
# Comandos para Gerenciamento de Processos

- kill
- Finaliza um job;
- Kill %id
  - %id é o número do job;
- jobs
- Exibe os jobs em execução pelo shell;
- \$ jobs [opções]
  - -l exibe o nome e o número de cada processo
  - -s exibe o nome de cada processo
  - -p exibe o número de cada processo



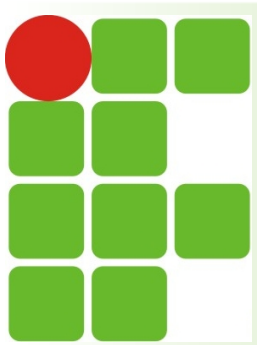
# Comandos para Gerenciamento de Processos

- ps
- Exibe informações sobre os processos ativos;
- ps [opções]
  - -a exibe informações de outros usuários
  - -u exibe o nome do usuário e a hora de início
  - -x exibe processos não associados ao terminal
  - -p PID exibe o processo cujo número é PID;
  - -l exibe linhas detalhadas
  - -e exibe todos os processos ativos
- \$ps -aux



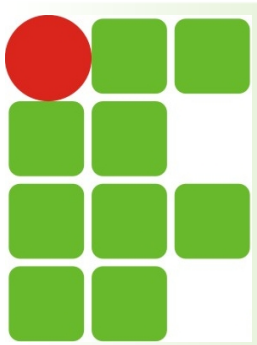
# Comandos para Gerenciamento de Processos

- Comando `ps -el` é exibida uma saída detalhada de todos os processos, inclusive com o estado do mesmo;
- S : sleeping (sem atividade);
- R : running (executando);
- D : waiting (aguarda um dispositivo de I/O);
- T : gestopt (suspenso, parado);
- Z : zombie(orfão);



# Comandos para Gerenciamento de Processos

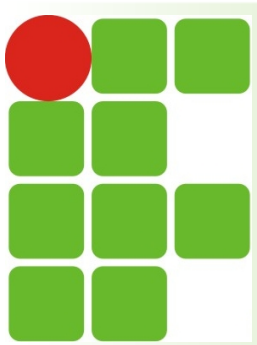
- `ps tree`
- Exibe informações sobre os processos ativos em forma de árvore (necessita de instalação do pacote);
- `$ps treee [opções]`
  - `-a` exibe argumentos de linha de comando
  - `-c` não compacta subárvores
  - `-l` exibe linhas detalhadas
  - `-n` classifica processos pelo id
  - `-p` mostra ids de processo



# Comandos para o Gerenciamento de Processos

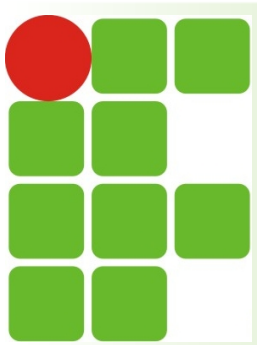
- `pidof processo`
- Retorna o PID do processo informado;
- `$pidof bash;`
- `top [opções]`
- Exibe os processos com maior consumo de CPU;
  - `-u usuário` Exibe apenas processos pertencentes ao usuário especificado
  - `-U` não mostra processos do usuário especificado
  - `-d n` Atualiza o monitor a cada `n` segundos;
  - `-t tty` Exibe processos apenas do terminal especificado;





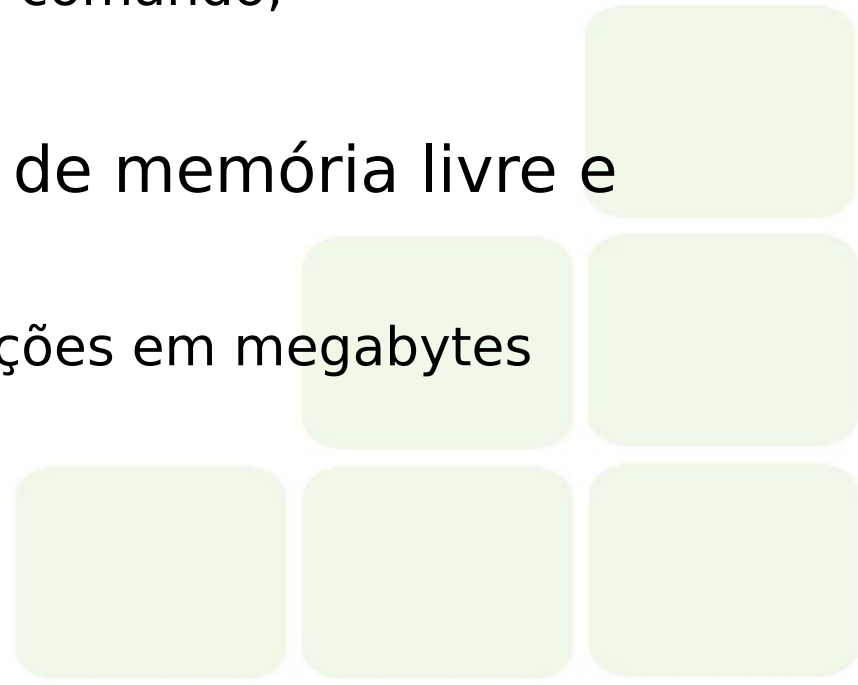
# Comandos para Gerenciamento de Processos

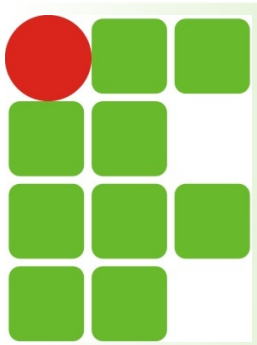
- tload [opções]
- Representa de forma gráfica a carga do sistema;
- Opções
  - -s número Mostra uma escala vertical com espaçamento especificado por número(entre 1 e 10)
  - -d número Especifica o intervalo entre atualizações em segundos;



# Comandos para Gerenciamento de Processos

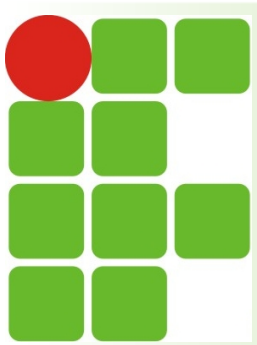
- uptime [-V]
  - Exibe o tempo de funcionamento do sistema e a sua carga;
    - -V Exibe a versão do comando;
- free [opções]
  - Exibe a quantidade de memória livre e usada no sistema;
    - -m Exibe as informações em megabytes





# Comandos para Gerenciamento de Processos

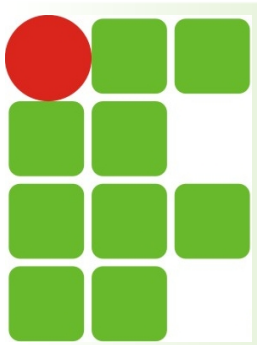
- `vmstat`
- Exibe as estatísticas do uso da memória virtual;
- `nice [opção] prioridade comando`
- Executa um processo com uma prioridade diferente da padrão;
- Prioridade varia entre -20 e 19, quanto menor, maior a prioridade;



# Comandos para Gerenciamento de Processos

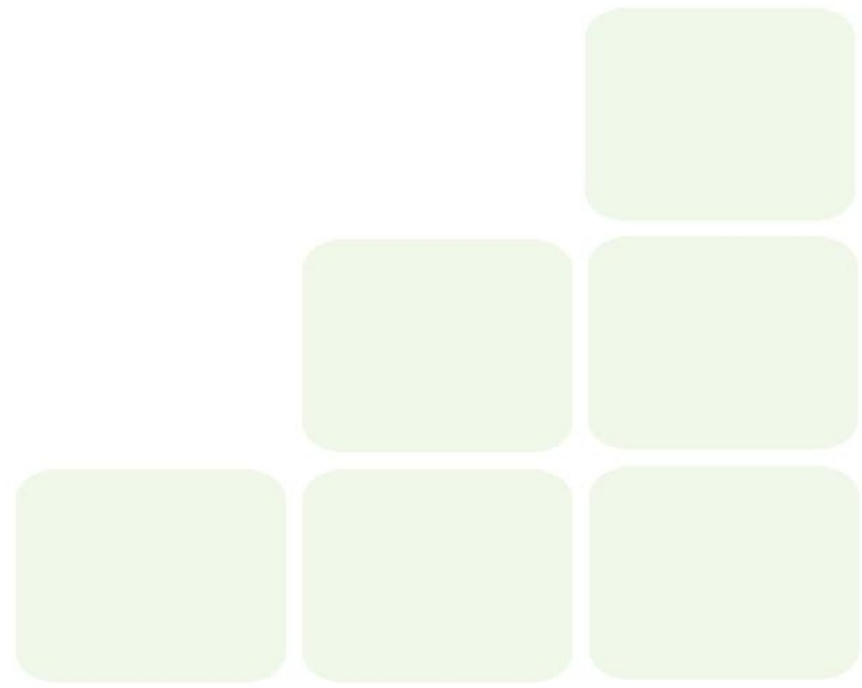
- renice prioridade [opções]
- Modifica a prioridade de um processo em execução;
- Pode ser aplicado a um processo, usuário ou grupo;
  - -p processo
  - -u usuário
  - -g grupo
- `$renice +15 1752`
- `$renice -2 -u usuario`
- `$renice +4 -g projeto`

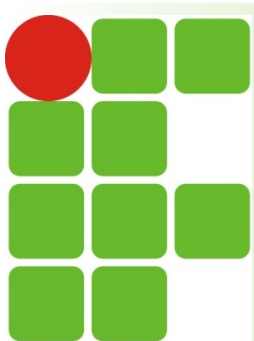




# Atividade

- Fazer a Atividade Prática IV





# Sistemas Operacionais de Redes

## ■ Bibliografia

- MORIMOTO, Carlos E.. Linux, Entendendo o Sistema – Guia Prático. Sul Editores, 2006.
- MORIMOTO, Carlos E.. Linux, Redes e Servidores – Guia Prático. Sul Editores, 2006.
- BATTISTI, Júlio. Windows Server 2003 Curso Completo. Axcel, 2003.
- THOMPSON, Marco Aurélio. Windows Server 2003 - administração de redes. Érica, 2003.