

Gerenciamento de Processos

- Um Processo é um Programa em Execução com todos os recursos necessários para este fim.
- Cada Processo, quando em execução, enxergar como sendo possuidor de todos os recursos do sistema.
- O Linux é um Sistema Preemptivo, isto quer dizer que um processo entra e sai do modo de execução a qualquer momento.
- O Agendado de Tarefas ou Scheduler de Sistema Operacionais modernos geralmente possuem três estados: Pronto, Em Execução e Em Espera (Bloqueado).

Gerenciamento de Processos

- Ao se Executar um Programa, este entra na Fila (queue) de execução. Cada Processo possui uma fatia (slice) de tempo para ser executado, terminada esta ele adormece volta para fila de processo e aguarda sua vez.
- O Processo pode bloquear (em espera) caso faça acesso a algum dispositivo de I/O (E/S).
- Ao se terminar a execução de um processo ele é retirado da fila de execução e todos os recursos alocados a ele são liberados.

Gerenciamento de Processos

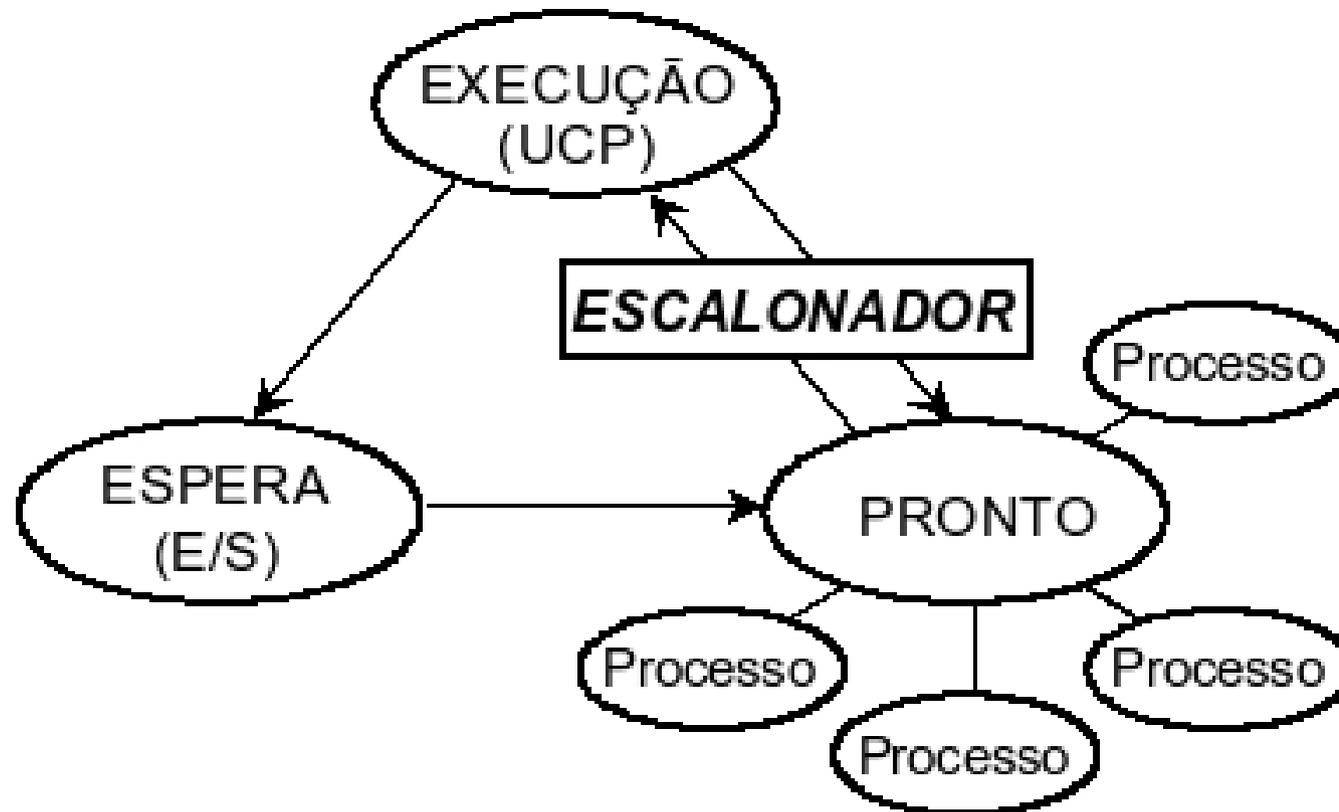


Imagem Retirada em: <http://lrodrigo.lncc.br>

Gerenciamento de Processos

- Um processo pode criar filhos. Um processo que cria filhos é chamado processo pai e os filhos são threads ou processo filhos.
- **Threads** são linhas de execução do processo pai, e estas compartilham com o processo pai arquivos e memória.
- O Linux trata processos e threads como iguais, o thread é considerado um processo que compartilha recursos com o processo que o criou.
- Processos filhos são mortos quando o processo pai é morto. Processos filhos que não finalizam quando o processo pai é finalizado são chamados de processos zombie.

Gerenciamento de Processos

- Podemos Iniciar, Pausar, Retomar, Recarregar, parar, matar priorizar, sinalizar, consultar e jogar em segundo plano um processo.
- Para conversar com um Processo nos sinalizamos o processo.
- Cada Processo possui um número identificador único chamado PID (Process ID).
- Para visualizar todos os processos em execução podemos utilizar por exemplo o comando 'ps aux'
- O kernel cria um diretório para cada processo sobre o diretório virtual /proc. Verifique listando todos os arquivos do diretório /proc.

Gerenciamento de Processos

Principais Comandos para Gerenciamento de Processos.

Consulta

- `ps`, `pstree`, `top`, `fuser`, `pidof` e `lsof`

Sinalização

- `kill` e `killall`.

Prioridade

- `nice` e `renice`

Segundo Plano

- `fg`, `bg` e `jobs` (plano de execução)

Gerenciamento de Processos

- Para gerenciar processos geralmente listamos e consultamos seu PID e então executamos alguma ação sobre ele através de sinalização.
- Cada Processo criado possui um diretório no /proc com informações a respeito dos recursos alocados para este processo.
- O processo de PID 1 é o primeiro processo de inicialização, geralmente é o comando init.
- init é o pai de todos os processos do sistema. Um PID é atribuído dinamicamente a um processo e é único durante a “vida” do sistema.

Gerenciamento de Processos

- ps

Lista os processos do sistema trazendo informações sobre seu estado (PID, tempo de execução, estado, nome do usuário owner, consumo de CPU e memória.

Ex.:

ps aux – todos os processos

ps l – outras informações

ps f – árvore de processos

ps -o pid,user,comm

ps -C nome_de_um_comando_a_ser_monitorado

ps -user root – lista processo de um usuário

Gerenciamento de Processos

Estados dos Processos (stats) (ps a)

- S – Em Espera (Sleeping)
- D – Lista processos blocked
- R – Fila de execução
- T - Parado
- W - Paginando
- X - Morto
- Z – Zumbi

Gerenciamento de Processos

Prática

Altere o console corrente, logue no sistema com uma conta que não seja o root, execute o editor vi/vim (vi) e volte para o console anterior (alt+f1).

- Liste todos os processos e encontre nesta listagem o Processo do usuário que executou o vi.
- Liste apenas o processo vi.
- Liste todos os processos apenas do usuário que iniciou o vi.
- Liste apenas o PID de todos processos.

Gerenciamento de Processos

- `ps`

Árvore de Processos – visualizar relação de processos pai e filho.

Opções:

- p – Mostra PID

- a – Linha de Comando Completa do Processo

Ex.:

```
ps -p
```

```
ps -a
```

Gerenciamento de Processos

- **top**

Monitoração interativa de processos.

- Opções:

- d – define intervalo de atualização

- u – monitorar processos de um determinado usuário

- **Alguns comandos de Utilização:**

- <Enter>** - Atualiza Lista de Processos

- P** – Ordenar pelo Uso de CPU

- M** – Ordenar pelo uso de Memória

- d** – Alterar Intervalo de Atualização

- u** – Monitra processo de um usuário

- r** – Altera Prioridade (renice)

- k** – Mata Processo (kill)

Gerenciamento de Processos

Prática

Liste todos os processos no formato de árvore e redirecione o resultado para o arquivo /tmp/lista_processos_arvore

Com o top mostrar o status dos processos em um intervalo de 1s (um segundo) ordenado pelo uso do Processador.

Iniciar o vi em outro Console (alt+f2) e verificar o PID deste, e matar (kill) este processo no primeiro console (alt+f1).

Monitorar apenas os processos do usuário root.

Gerenciamento de Processos

- pidof

Mostra o PID do processo correspondendo (nome).

-x Pesquisa também nomes de Shell scripts

```
Ex.: pidof init
      pidof login
      pidof udevd
      pidof vi
```

- lsof

Lista arquivos abertos e os processos donos destes

```
Ex.: lsof
```

Gerenciamento de Processos

- kill (sinalizador de processos)

Sinaliza um Processo. Geralmente utilizado para “matar” um processo. O Sinal padrão do kill é o 15 (SIGTERM) que é uma forma “educada” de pedir ao processo que termine.

O Extremo do kill é o kill -9, onde o processo é “morto a queima roupa”. O Processo é finalizado e não pode salvar arquivos que esta trabalhando

Opções:

-l – lista todos os sinais

Ex.:

```
kill -l
```

Gerenciamento de Processos

- `killall`

Envia Sinalização a um ou mais processos de mesmo nome.

- i modo interativo

- v mostra mensagens (verbose mode)

- w esperar os processo finalizarem.

Gerenciamento de Processos

Altere o console corrente, logue no sistema com uma conta que não seja o root, execute o editor vi e volte para o console anterior.

- Liste todos os processos e encontre o Processo do usuário que executou o vi.
- Sinalize para que o processo vi criado seja terminado “educadamente). Utilize o comando kill.
- repita a operação de executar o editor vi, busque seu PID e sinalize para que este seja morto a “queima roupa”. Utilize o comando kill.

Gerenciamento de Processos

- nice

Inicia o Programa com a Prioridade especificada. Um Programa com prioridade máxima possui mais tempo de processamento.

Opções:

-n Valor da prioridade

Prioridade:

-20 Prioridade Maxima

-10 Prioridade Alta

0 Prioridade Normal

10 Prioridade Baixa

19 Prioridade minima

Ex.:

```
nice -n -19 vi
```

Gerenciamento de Processos

- `renice`

Modificar a prioridade em tempo-real

- p faixa de valor de PID

- u processos de um usuario especifico

Ex.: `renice 20 PID DO PROCESSO`

Gerenciamento de Processos

Prática

- nice e **renice**

Inicie o vi com prioridade mínima em um outro console (alt+f2), usando o **top** verifique apenas a prioridade deste processo, modifique a prioridade deste para máxima e finalize este sinalizando com SIGTERM (15) utilizando o comando **top** ou **kill**.

Inicie o vi com prioridade máxima em um outro console (alt+f2), usando o **ps** verifique apenas a prioridade deste processo.

Gerenciamento de Processos

- fg, bg e jobs

Com os comandos fg e bg é possível colocar um processo em segundo plano (plano de execução)

Para isto, inicie um programa e pare este com SIGSTOP (CTRL+Z), execute o bg para que o processo continue sua execução em segundo plano.

Para trazer este para primeiro plano utilize o fg.

Experimente fazer isto com o comando nano

Outra forma de Executar um comando em segundo plano é utilizar &

Ex.: vi &

ls /bin &

Gerenciamento de Processos

Prática

- Liste todos os processos do usuário **root** mostrando apenas as informações de **pid, usuário e comando**. Direcione a saída para o arquivo **/tmp/revisao2_processo**.
- Mostre todos os processos em formato de árvore.
- Liste todos os processos que estão utilizando o arquivo **/bin/bash** e direcione (concatenando) o resultado para o arquivo **/tmp/revisao2_processo**.
- Usando o **top** visualize todos os processos em um taxa de atualização de 10s (dez segundos) e ordenado pelo uso de CPU. Altere a prioridade do processo que esta no topo desta lista para Máxima.

Gerenciamento de Serviços

- Um serviço é um programa especial que provê determinadas funções ao sistema.
- Serviços também são chamados de “daemons” e são executados em segundo plano (background).
- Serviços em geral esperam por requisições em portas especificadas nas RFC.
- Exemplos de Serviços são o Serviço de Mouse (GPM), Crontab (Agendamento de Tarefas, Servidores de FTP, Web (Apache), POP3 (Sendmail), Gerenciamento Log (syslog), Sistemas Gerenciadores de Banco de Dados (MySQL, PostgreSQL, Oracle e etc.).
- Serviços podem ser dependentes ou independentes de um servidor de serviços (inetd).

Gerenciamento de Serviços

- Cada serviço possui um script para sua gerencia (initscripts) e inicialização.
- Estes initscripts geralmente residem no diretório /etc/init.d. (ex.: `sh /etc/init.d/cron`)
- Existe um comando para facilitar a gerencia dos scripts em Distribuições Red Hat-Like como o CentOS e Ubuntu. O comando é o service.

Ex.: `service cron`

Ações Comuns: `start`
`stop`
`status`
`reload`

Gerenciamento de Serviços

- Ações sobre Serviços

`start` – inicia o serviço

`stop` – parar o serviço

`status` – reportar o status

`restart` – reinicia o serviço

`reload` – reinicia o serviço sem tirá-lo de execução

Ex.: `service cron status`

`service cron stop`

`service cron start`

`service cron reload`