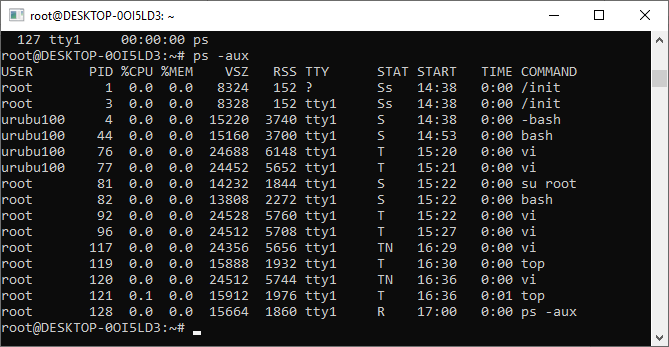
***4. Gerenciamento de processos e tarefas***

4.3 Ciclo de vida processos – gestão e hierarquia

4.4 Threads: conceitos, modelos e programação

Antes de iniciarmos vamos revisar as repostas da atividade aula anterior item 5, a e b.

1. Na tela abaixo o que significa Ss e TN na coluna STAT, explique



ESTADOS DOS PROCESSOS

D Uninterruptible sleep (usually IO) – processo em modo sleepping initerrupto (em geral relativos a E/S) – sem interrupção contínua.

R Running or runnable (on run queue) – rodando ou em execução (na fila de execução).

**S Interruptible sleep (waiting for an event to complete) – Interrupção momentânea ( em geral enquanto aguarda a conclusão de um evento.**

**T Stopped, either by a job control signal or because it is being traced – interrompido por um sinal de controle ou por causa de algo que é rastreado.**

X dead (should never be seen) - morto

Z Defunct ("zombie") process, terminated but not reaped by its parent. Processo morto, relativo ao processo pai.

Os subcaracteres são:

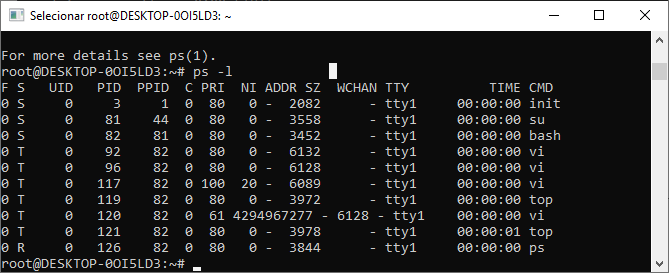
**N low-priority (nice to other users) – baixa prioridade, fornecendo-a a processos de outros usuários**

L has pages locked into memory (for real-time and custom IO) – mostra que há páginas bloqueadas na memória.

**s is a session leader – mostra se a sessão é líder. Exemplo do shell ou de um bash. Ambos são pai porque podem ser executados por diferentes grupos de usuários. Neste caso a sessão é o líder, se kill na sessão, kill em todos os processos filhos.**

l is multi-threaded – processos multiencadeados

1. O que significa ADDR SZ e WCHAN na tela abaixo, explique



%CPU - Quanto da CPU o processo está usando

%MEM - Quanta memória o processo está usando

ADDR - Endereço de memória do processo

C ou CP - Informações de uso e agendamento da CPU

CMD - Nome do processo, incluindo argumentos, se houver

NI – nice

PID - Número de identificação do processo

PPID - Número de identificação do processo pai do processo

PRI - Prioridade do processo

TIME – tempo de uso total da CPU

TT ou TTY - Terminal associado ao processo

WCHAN (Waiting Channel) – (rotina do kernel para processo waiting relativo ao endereço de memória) Endereço de memória do evento pelo qual o processo está aguardando, processos em execução são marcados por um hífém ( não há endereço fixo na memória para indicar)

Voltando ao conteúdo:

4.3 Ciclo de vida processos – gestão e hierarquia

4.4 Threads: conceitos, modelos e programação

**Composição de um processo**

O sistema operacional lida com uma infinidade de processos , sendo necessário meios de controle. Os processos apresentam um conjunto de características:

- Proprietário/grupo do processo;  
- Estado do processo (em espera, em execução, etc);  
- Prioridade de execução;  
- Recursos de memória.

**Grupo de processos**

• Compartilhamento de recursos – Baseados em hierarquia de processos:

• Um processo pai cria processos filhos;

• Os filhos podem executar o mesmo código, ou trocá-lo;

• Obtem-se uma árvore de processos.

• Implica na definição da semântica de término de um processo:

– Só o processo morre;

– Toda sua descendência morre.

Os processos devem interagir com o disco para armazenar e recuperar dados não voláteis. O disco físico é abstraído pelo Sistemas de Arquivos, de acordo com uma hierarquia: – Diretórios – Arquivos.

Os diretórios estão freqüentemente organizados de acordo com uma hierarquia em árvore – Raiz (‘/’)

- Diretório de trabalho de um processo ( ‘.’)

– Caminho relativo / absoluto

- Deve ter chamadas de sistema para acessar o Sistema de Arquivos.

Exemplos de chamadas de sistema:

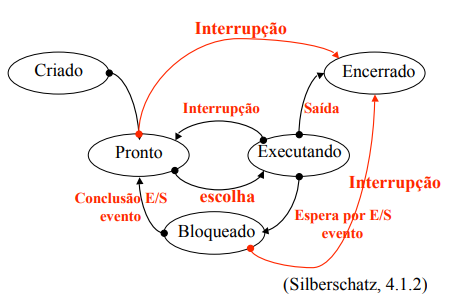
Relativas ao Gerenciamento de processos: – fork(), waitpid(), exit(), execve(...), getpid()...

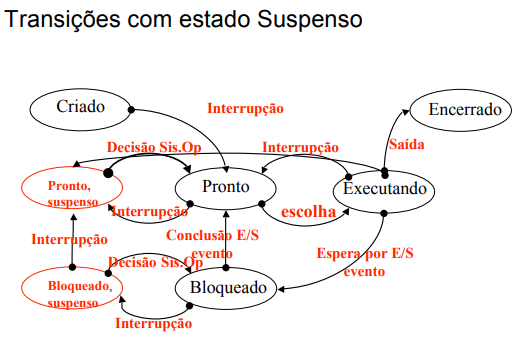
Relativas aos Sinais – sigaction(), sigreturn(), sigprocmask(), kill()...

Relativas ao Gerenciamento de arquivos – open(), close(), mknod(), read(), write(), pipe(),... – mkdir(), mount(),... Relativas aos Direitos de acesso

Relativas ao Gerenciamento de tempo – Time()

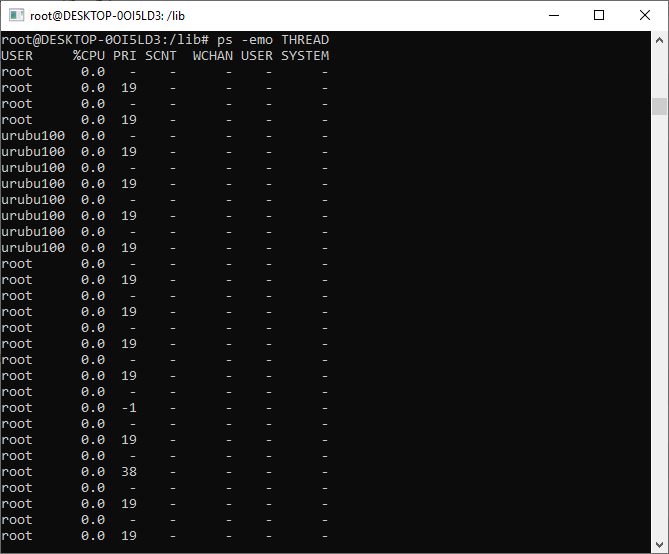
DIAGRAMA DE ESTADOS





Agora vamos analisar as sáidas do comando relativo as threads

ps -emo THREAD



Todas as saídas em hífem significam que estão em execução.

PRI – prioridade

SCNT – switch count

O kernel alterna entre threads em um esforço para compartilhar a *CPU* efetivamente; essa atividade é chamada de *alternância de contexto* . Quando um thread é executado pela duração de seu intervalo de tempo ou quando é bloqueado porque requer um recurso indisponível no momento, o kernel encontra outro thread para executar e o contexto muda para ele. O sistema também pode interromper o thread em execução no momento para executar um thread acionado por um evento assíncrono, como uma interrupção do dispositivo. Embora os dois cenários envolvam a alternância do contexto de execução da *CPU* , a alternância entre threads ocorre de forma *síncrona* em relação ao segmento atualmente em execução, enquanto as interrupções de manutenção ocorrem de maneira *assíncrona* em relação ao encadeamento atual. Além disso, as alternâncias de contexto entre processos são classificadas como *voluntárias* ou *involuntárias.*Uma opção de contexto voluntária ocorre quando um encadeamento é bloqueado porque requer um recurso indisponível. Uma alternância involuntária de contexto ocorre quando um encadeamento é executado pelo período de tempo ou quando o sistema identifica um encadeamento de prioridade mais alta a ser executado.

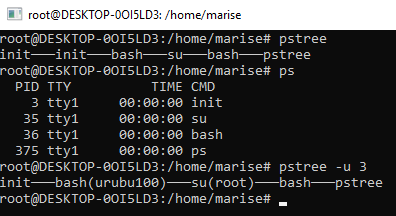
Cada tipo de alternância de contexto é feito através de uma interface diferente. A troca de contexto voluntária é iniciada com uma chamada para a rotina *sleep* (), enquanto uma troca de contexto involuntária é forçada pela invocação direta do mecanismo de troca de contexto de baixo nível incorporado nas *rotinas mi\_switch* () e *setrunnable* (). A manipulação de eventos assíncronos é acionada pelo hardware subjacente e é efetivamente transparente para o sistema. Nossa discussão se concentrará em como o tratamento de eventos assíncronos se relaciona à sincronização do acesso às estruturas de dados do kernel.

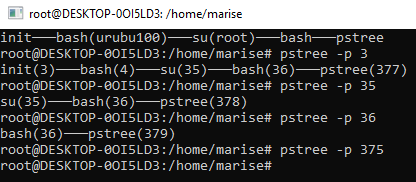
**pstree:** esse comando mostra processos relacionados em formato de árvore. Sua sintaxe é:

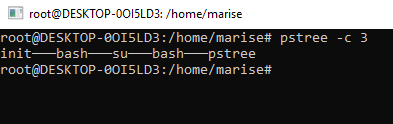
**pstree -opção PID**

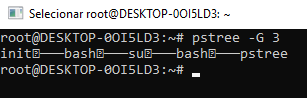
Entre as opções, tem-se:

**-u -** mostra o proprietário do processo;  
**-p -**exibe o PID após o nome do processo;  
**-c -**mostra a relação de processos ativos;  
**-G -** usa determinados caracteres para exibir o resultado em um formato gráfico.



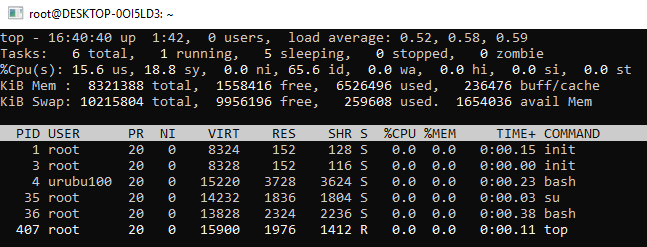




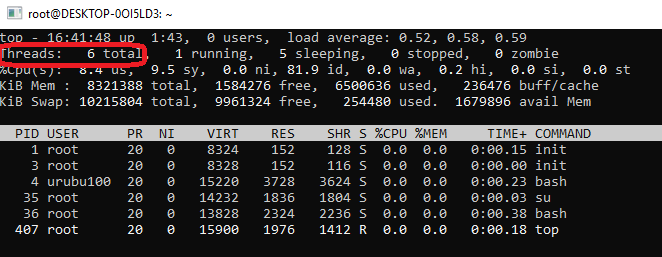


Agora vamos visualizar as Threads:

Qdo vc der o comando top:



Agora aperte shift +h



Em Linux as threads são entendidas como as tasks

Agora vamos a nossa atividade de fixação de conceitos do conteúdo abordados sobre Gerenciamento de Processos. Esta atividade pontua se entregue até as 21 horas. Faça Upload das respostas em um arquivo separado deste. Pode copiar as perguntas e colocar as respostas abaixo de cada uma delas. Estas questões são associadas a certificações.

ATIVIDADE DA AULA DE 26/05/2020 – PONTUA ENTREGA NA DATA DE HOJE

NOME DO ALUNO Mike Melo Azevedo

RA 01192122

1. O Sistema Operacional LINUX, na sua configuração padrão, é uma alternativa ao uso do Sistema Operacional Windows. Ele possui, entre outras características.

***a) multitarefa, memória virtual, biblioteca compartilhada, gerenciamento de memória próprio e rede TCP/IP.***

b) servidor IIS capaz de hospedar e executar páginas ASP.

c) sistema de arquivo NTFS, FAT e FAT 32.

d) Active Directory.

e) servidores DNS e WINS.

2) Analise as seguintes afirmações relativas à liberdade dos usuários de um Software livre.

1. A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades, exceto alteração no código-fonte. II.
2. A liberdade de executar o programa, para qualquer propósito.
3. A liberdade de utilizar cópias de modo que se possa ajudar outros usuários, sendo vedada a redistribuição.
4. Aquele que redistribuir um software GNU poderá cobrar pelo ato de transferir uma cópia ou poderá distribuí-las gratuitamente. Indique a opção que contenha todas as afirmações verdadeiras.

a) I e II

b) I e III

c) III e IV

***d somente I***

e) II e IV

3) Os programas, normalmente instalados no disco rígido, que permitem ao usuário escolher entre dois ou mais sistemas operacionais instalados na máquina são conhecidos como gerenciadores de boot. Um dos mais comuns gerenciadores de boot para ambiente linux é o:

***a) GRUB***

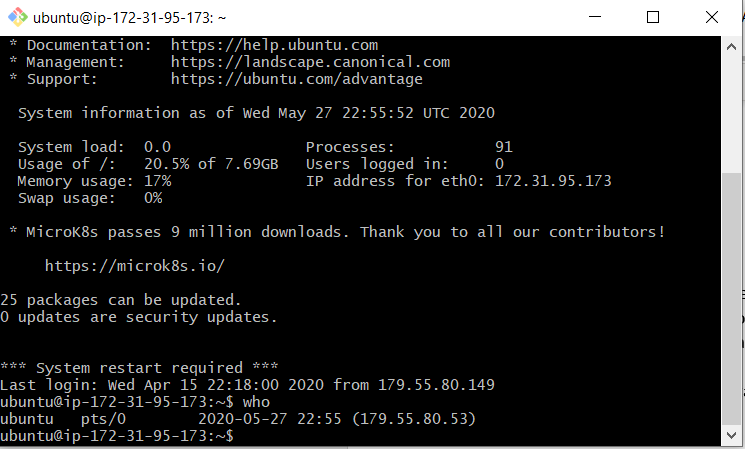
b) Kde

c) gnome

d) conectiva

e) redhat

4) Abra um terminal Linux. Solicite a informação sobre quem está trabalhando nesta máquina (o Linux é um sistema multiusuário e multitarefa): execute o comando "who". Qual o resultado? Mostre o print da tela com resultado.



5) Um processo é uma instância, e uma instância é uma Thread em máquinas na nuvem. Qual é a relação disso?

**As Threads podem compartilhar a utilização de um recurso, mas este compartilhamento ocorre apenas dentro de um único processo, pois o compartilhamento entre processos não existe.**

6) O que o comando top executa e qual a associação com status running e sleeping?

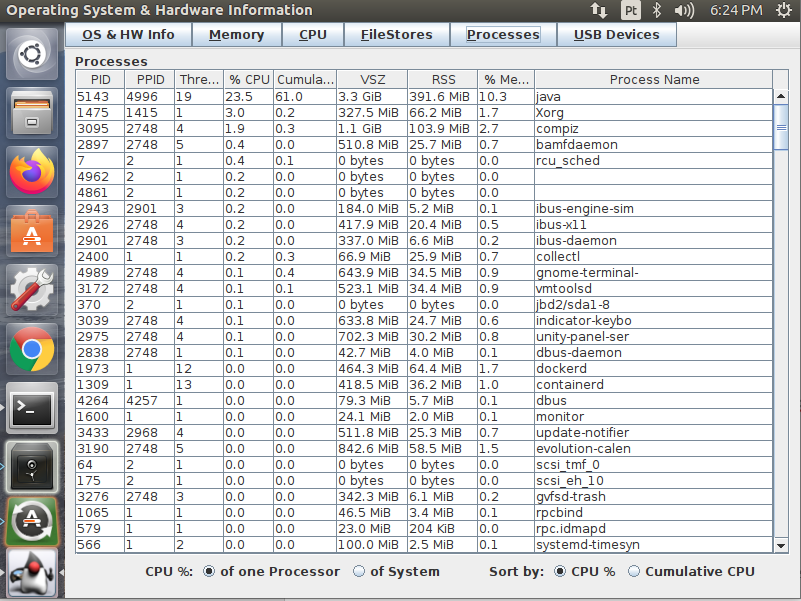
**O comando top ele mostra os processos ativos no sistema Linux e ele consegue verificar os status dos processos se estão rodando ou se estão descanso.**

7) Quais os recursos de hardware que o comando top apresenta? .

**Percentual de uso de CPU e memória**.

8) O que é um PID e um PPID? Cite um exemplo e apresente um print de tela com esse exemplo.

**PID e PPID são processos, aonde o PID é o processo em si e PPID é o processo Pai.**



9) Qual a diferença entre o comando job e o ps, qual a finalidade de cada um.

10) O Sistema Operacional Linux é muito utilizado em servidores, e o uso é realizado muitas vezes por meio do terminal via comando. Existem diversos comandos por meio de terminal para que se possa verificar o seu desempenho através de um monitoramento. Qual o comando que exibe as tarefas do Linux?

**top**