Relembrando

Em nossas primeiras aulas discutimos o conceito do que seria um processo!

Quem lembra?

Processos

- Essencialmente podemos dizer que é uma aplicação em execução.
- Um processo consome memória, disco, CPU.
- O sistema operacional aloca recursos para cada processo compartilhando-os no tempo e no espaço
- Escalonador de processos!

Processos

- O SO lida com vários processos e precisa ter como controlá-los (gerenciamento de processos)
- Para isso os processos podem ser caracterizados:
  - Proprietário do processo;
  - Estado do processo (em espera, em execução, etc.);
  - Prioridade de execução;
  - Consumo de recursos de memória
  - Consumo de processamento

Processos

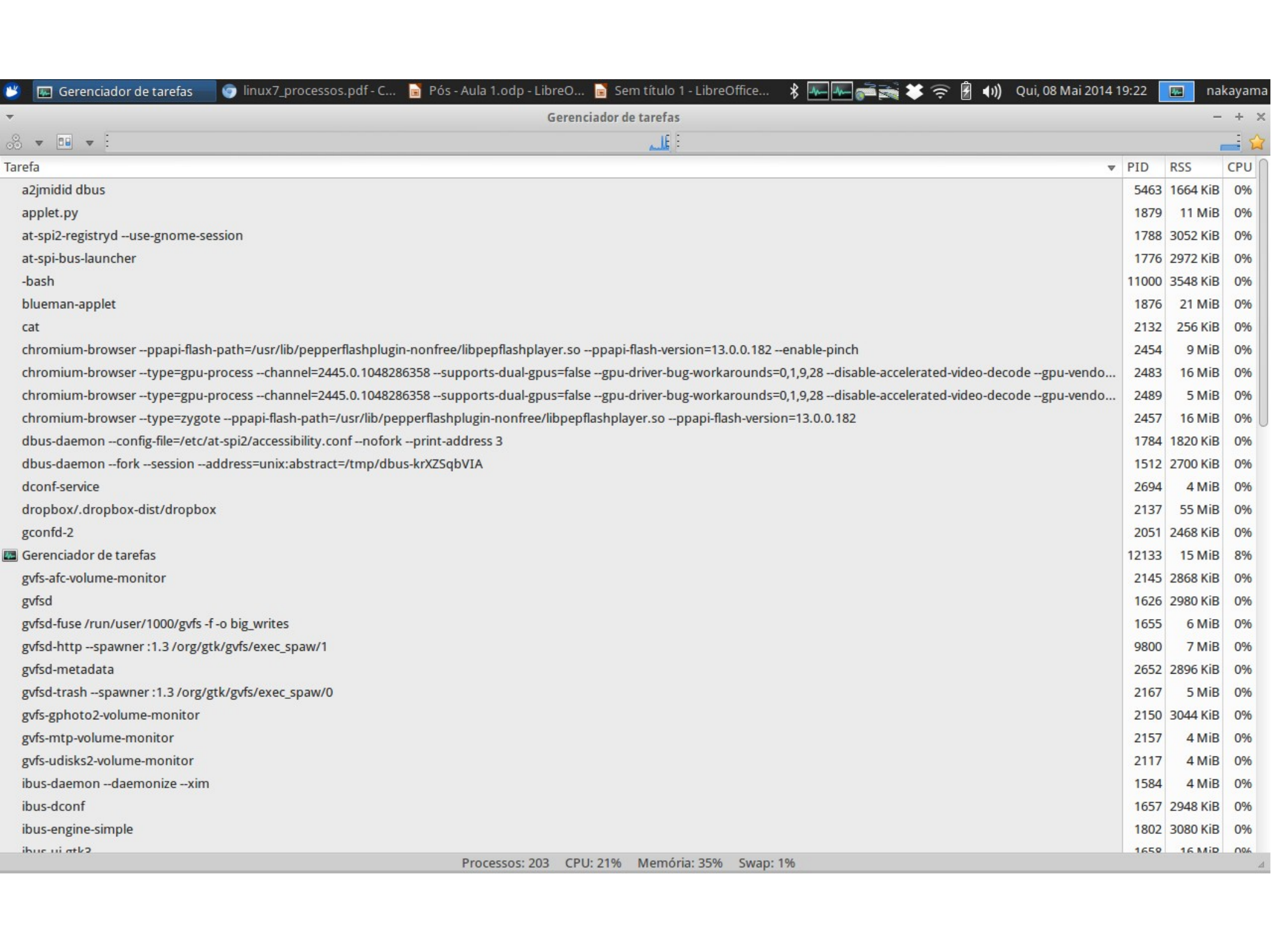
- Para saber quem é o dono do processo, e assim verificar suas permissões, cada processo está associado ao UID e GID do usuário que o criou
- Mensagens especiais chamadas de sinais são utilizadas para permitir a comunicação entre processos e para que o SO possa interferir em um processo

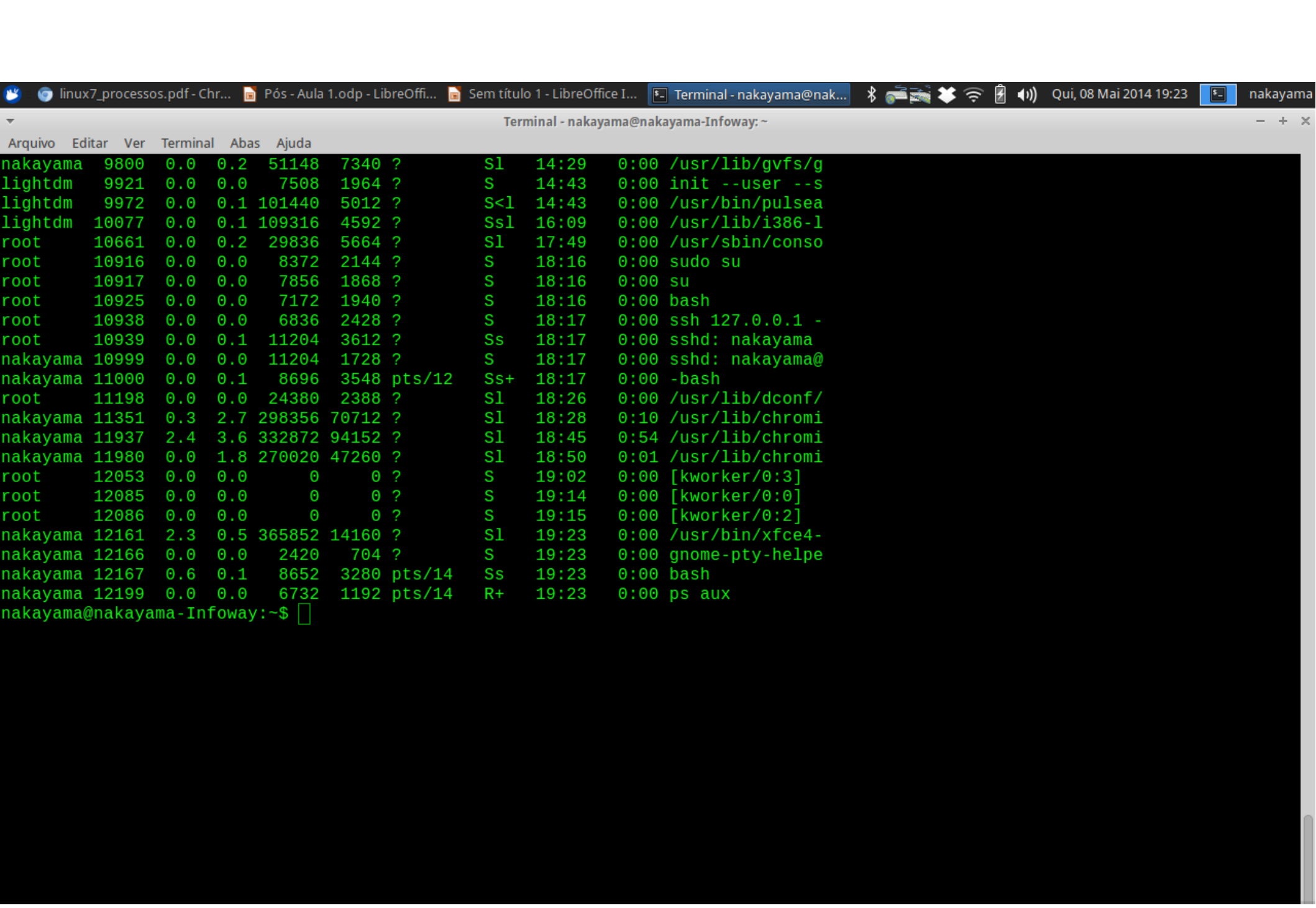
Processos

- Alguns sinais
- STOP - esse sinal tem a função de interromper a execução de um processo e só reativá-lo após o recebimento do sinal CONT;
- CONT - esse sinal tem a função de instruir a execução de um processo após este ter sido interrompido;
- SEGV - esse sinal informa erros de endereços de memória;
- TERM - esse sinal tem a função de terminar completamente o processo, ou seja, este deixa de existir após a finalização;
- ILL - esse sinal informa erros de instrução ilegal, por exemplo, quando ocorre divisão por zero;
- KILL - esse sinal tem a função de "matar" um processo e é usado em momentos de criticidade.

Processos

- Verificação em modo gráfico ou em terminal!


Gerenciador de tarefas
linux7_processos.pdf - C...
Pós - Aula 1.odp - LibreO...
Sem título 1 - LibreOffice...
Qui, 08 Mai 2014 19:22
nakayama
Gerenciador de tarefas
Tarefa
PID
RSS
CPU
a2jmidid dbus 5463 1664 KiB 0%
applet.py 1879 11 MiB 0%
at-spi2-registryd --use-gnome-session 1788 3052 KiB 0%
at-spi-bus-launcher 1776 2972 KiB 0%
-bash 11000 3548 KiB 0%
blueman-applet 1876 21 MiB 0%
cat 2132 256 KiB 0%
chromium-browser --ppapi-flash-path=/usr/lib/pepperflashplugin-nonfree/libpepflashplayer.so --ppapi-flash-version=13.0.0.182 --enable-pinch 2454 9 MiB 0%
chromium-browser --type=gpu-process --channel=2445.0.1048286358 --supports-dual-gpus=false --gpu-driver-bug-workarounds=0,1,9,28 --disable-accelerated-video-decode --gpu-vendo... 2483 16 MiB 0%
chromium-browser --type=gpu-process --channel=2445.0.1048286358 --supports-dual-gpus=false --gpu-driver-bug-workarounds=0,1,9,28 --disable-accelerated-video-decode --gpu-vendo... 2489 5 MiB 0%
chromium-browser --type=zygote --ppapi-flash-path=/usr/lib/pepperflashplugin-nonfree/libpepflashplayer.so --ppapi-flash-version=13.0.0.182 2457 16 MiB 0%
dbus-daemon --config-file=/etc/at-spi2/accessibility.conf --nofork --print-address 3 1784 1820 KiB 0%
dbus-daemon --fork --session --address=unix:abstract=/tmp/dbus-krXZSqbVIA 1512 2700 KiB 0%
dconf-service 2694 4 MiB 0%
dropbox/.dropbox-dist/dropbox 2137 55 MiB 0%
gconfd-2 2051 2468 KiB 0%
Gerenciador de tarefas 12133 15 MiB 8%
gvfs-afc-volume-monitor 2145 2868 KiB 0%
gvfsd 1626 2980 KiB 0%
gvfsd-fuse /run/user/1000/gvfs -f -o big_writes 1655 6 MiB 0%
gvfsd-http --spawner :1.3 /org/gtk/gvfs/exec_spaw/1 9800 7 MiB 0%
gvfsd-metadata 2652 2896 KiB 0%
gvfsd-trash --spawner :1.3 /org/gtk/gvfs/exec_spaw/0 2167 5 MiB 0%
gvfs-gphoto2-volume-monitor 2150 3044 KiB 0%
gvfs-mtp-volume-monitor 2157 4 MiB 0%
gvfs-udisks2-volume-monitor 2117 4 MiB 0%
ibus-daemon --daemonize --xim 1584 4 MiB 0%
ibus-dconf 1657 2948 KiB 0%
ibus-engine-simple 1802 3080 KiB 0%
Processos: 203 CPU: 21% Memória: 35% Swap: 1%


linux7_processos.pdf - Chr...
Pós - Aula 1.odp - LibreOffi...
Sem título 1 - LibreOffice I...
Terminal - nakayama@nak...
Qui, 08 Mai 2014 19:23
nakayama
Terminal - nakayama@nakayama-Infoway: ~
Arquivo Editar Ver Terminal Abas Ajuda
nakayama 9800 0.0 0.2 51148 7340 ? Sl 14:29 0:00 /usr/lib/gvfs/g
lightdm 9921 0.0 0.0 7508 1964 ? S 14:43 0:00 init --user --s
lightdm 9972 0.0 0.1 101440 5012 ? S<l 14:43 0:00 /usr/bin/pulsea
lightdm 10077 0.0 0.1 109316 4592 ? Ssl 16:09 0:00 /usr/lib/i386-l
root 10661 0.0 0.2 29836 5664 ? Sl 17:49 0:00 /usr/sbin/conso
root 10916 0.0 0.0 8372 2144 ? S 18:16 0:00 sudo su
root 10917 0.0 0.0 7856 1868 ? S 18:16 0:00 su
root 10925 0.0 0.0 7172 1940 ? S 18:16 0:00 bash
root 10938 0.0 0.0 6836 2428 ? S 18:17 0:00 ssh 127.0.0.1 -
root 10939 0.0 0.1 11204 3612 ? Ss 18:17 0:00 sshd: nakayama
nakayama 10999 0.0 0.0 11204 1728 ? S 18:17 0:00 sshd: nakayama@
nakayama 11000 0.0 0.1 8696 3548 pts/12 Ss+ 18:17 0:00 -bash
root 11198 0.0 0.0 24380 2388 ? Sl 18:26 0:00 /usr/lib/dconf/
nakayama 11351 0.3 2.7 298356 70712 ? Sl 18:28 0:10 /usr/lib/chromi
nakayama 11937 2.4 3.6 332872 94152 ? Sl 18:45 0:54 /usr/lib/chromi
nakayama 11980 0.0 1.8 270020 47260 ? Sl 18:50 0:01 /usr/lib/chromi
root 12053 0.0 0.0 0 0 ? S 19:02 0:00 [kworker/0:3]
root 12085 0.0 0.0 0 0 ? S 19:14 0:00 [kworker/0:0]
root 12086 0.0 0.0 0 0 ? S 19:15 0:00 [kworker/0:2]
nakayama 12161 2.3 0.5 365852 14160 ? Sl 19:23 0:00 /usr/bin/xfce4-
nakayama 12166 0.0 0.0 2420 704 ? S 19:23 0:00 gnome-pty-helpe
nakayama 12167 0.6 0.1 8652 3280 pts/14 Ss 19:23 0:00 bash
nakayama 12199 0.0 0.0 6732 1192 pts/14 R+ 19:23 0:00 ps aux
nakayama@nakayama-Infoway:~$

Manipulação de processos

- Utilizado para enviar sinais aos processo
- Formato geral
- kill -SINAL <PID>
- Caso não seja especificado o sinal, o comando atua enviando um sinal TERM
- Como descobrir o PID? Utilizando o comando ps!

kill

- Exemplos

  kill -STOP 4220

- Interromper temporariamente a execução do processo 4220

  kill -CONT 4220

- Retomar a execução do processo 4220

  kill -STOP -1

- Envia o sinal de interrupção temporária para todos os processos

kill

- Exemplos
  - kill 4220
- Envia sinal TERM para o processo 4220. Este sinal pode ser ignorado pelo processo
  - kill -9 4220
- Envia sinal KILL para o processo 4220 forçando seu término

killall

- Mas, caso você esqueça disso e não saiba o PID do processo você pode utilizar o comando killall
- Formato geral
  - killall -SINAL <nome do processo>
- Existe algum problema em enviarmos um sinal ao processo pelo nome?
- Iremos enviar para todos os processos com aquele nome!

pstree

- Utilizado para visualizar a relação entre os processos que estão rodando no sistema!
- Formato
  - pstree [opções]
- Opções

  -a: apresenta argumentos e opções do comando

  -p: apresenta PID

nice e renice

- Processos possuem prioridades indicadas por valores inteiros entre -19 e 19
- Valores mais baixos indicam processos mais gulosos por recursos e valores mais altos indicam processos mais gentis!
- Valores negativos podem ser utilizados por usuário root!
- O Linux automaticamente determina as prioridades para os processos, porém podemos querer realizar algumas alterações.

nice

- Permite definir a prioridade de um processo a ser executado
- Ao executarmos o comando o processo é iniciado!
- Formato
  - nice [opções] <processo a executar>
- Opções

    -n: define a prioridade do processo

nice

- Exemplos

  nice -n -5 ntpd

  Define prioridade -5 para o processo ntpd

  nice -n -19 firefox

  Define prioridade -19 para o processo firefox

  nice -n 13 banshee

  Define prioridade 13 para o processo banshee

renice

- Para um processo que já está em execução usamos o comando renice
- Formato geral
  - renice <prioridade> <processo> [opções]
- Opções

  -u: a alteração ocorrerá nos processos do usuário informado

  -g: a alteração ocorrerá nos processos do grupo indicado;

  -p: a alteração ocorrerá no processo cujo PID for informado

  -n: valor da prioridade a ser utilizado

renice

- Exemplos
  - **renice 19 1000 -u aluno**
- Alterando prioridade do processo 1000 e dos processos do usuários aluno para 19
  - **renice 1 2824**
- Alterando prioridade do processo 2824 para 1
  - **renice 1 –p 1002**
- Alterando prioridade do processo 1002 para 1

Liberando o terminal

- Faça o teste, execute os comandos no terminal:
  - firefox
  - firefox&

- O que acontece com o terminal após o comando nos dois casos?

Colocando em background

- Execute o comando e utilize Ctrl z para colocar em background.
- Recupere o programa com bg na tela aonde o programa foi colocado em espera.

nohup

- Algumas vezes queremos iniciar um processo em uma máquina e desejamos que ele fique ativo mesmo após o usuário realizar logout
- Para isso utilizamos o comando nohup!
- Formato geral
  - nohup <comando a ser executado>

time

- Serve para contabilizar quanto tempo leva a execução de um comando

- Formato
  - time [opções] <comando a ser executado>:
    - apresenta o resultado do comando e a contagem de tempo
  - times [opções] <comando a ser executado>:
    - apresenta apenas a contagem de tempo

shutdown

- Usado para desligar a máquina
- Executado apenas por usuário root
- Formato
  - shutdown [opções] <tempo> <mensagem>
- Opções
  - -h: informa que o sistema deve ser desligado
  - -r: informa que o sistema deve ser reiniciado
  - -c: cancela um shutdown agendado

shutdown

- Exemplos
  - sudo shutdown -h 20
  - O sistema deve ser desligado em 20 minutos
  - sudo shutdown -r 20
  - O sistema deve ser reiniciado em 20 minutos
  - sudo shutdown -r 23:00
  - O sistema deve ser reiniciado às 23:00h
  - sudo shutdown –c
  - Cancela a operação agendada

shutdown

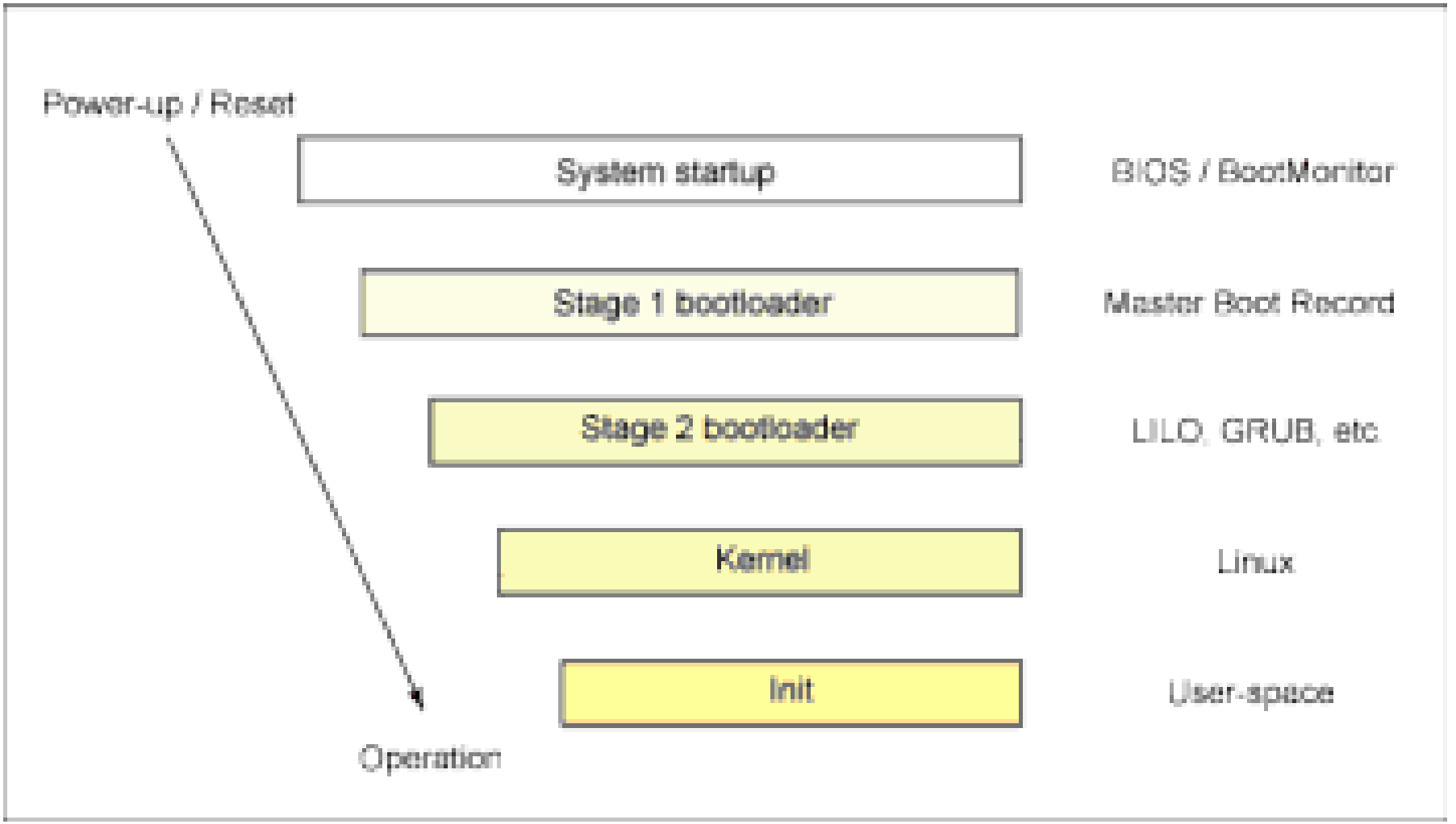
- Exemplos
  - sudo shutdown -h now
    - Desligue a máquina agora!
  - sudo shutdown -r now
    - Reinicie a máquina agora!

Processo de carga

- Processo pelo qual a máquina inicia e se prepara para ser utilizada!

Visão geral

- Tudo inicia através da execução de um código definido em um local conhecido
  - BIOS
- É feita a escolha de um dispositivo de boot
  - Loader de boot de 1 e 2 estágio são carregados
- Controle é passado ao kernel
- Primeiro programa de usuário (init) inicia o sistema em alto nível


Power-up / Reset
System startup
BIOS / BootMonitor
Stage 1 bootloader
Master Boot Record
Stage 2 bootloader
LILO, GRUB, etc.
Kernel
Linux
Init
User-space
Operation