

Sistemas Operacionais

Othon Oliveira

Fatec – Faculdade de Informática — PE

6 de fevereiro de 2016

Exemplos de sistemas operacionais



1- Programas ?

2- Processos ??

MS-Word, Google-Chrome, Firefox, Whatsapp,...



- Programas solicitam serviços ao sistema operacional através de **Chamadas de Sistemas**
- Programas bem escritos informam ao sistema operacional exatamente do que necessitam
- Na compilação são passadas informações geralmente “escondidas” nas bibliotecas do sistema, exemplo: `printf()` – linguagem C

Principais componentes do Kernel

A parte do sistema operacional responsável pelas chamadas de sistema é o núcleo ou *Kernel*

- gerência de processos *
- gerência de memória
- sistema de gerência de arquivos *filesystem*
- gerência de entrada e saída

Programas: conjunto de instruções que descrevem tarefas para serem executadas num computador

Processos: um programa sendo executado + dados necessários a sua execução
É de responsabilidade do sistema operacional gerir os processos do sistema de forma a maximizar a utilização da CPU.

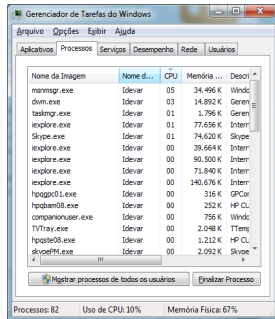
Processos

```

1 [|||||] 11.7% Tasks: 129, 308 thr; 1 running
2 [|||||] 6.9% Load average: 0.31 0.32 0.30
3 [|||||] 11.4% Uptime: 01:49:01
4 [|||||] 7.5%
Mem: ||||||| 10147/131040
Swap: 0/131040

PID USER      SS  RES  VSZ  %Mem CPU  %CPU  COMMAND
2727 root      Ss  409M 15445 8.6 1.3 142.17 /opt/google/chrome/chrome --type=process --channel=2088.2.721943447 --en
7218 root      Ss  113M 71612 5.7 1.4 132.73 compiz
1224 root      Ss  333M 4080 1.636 5 7.3 9.9 2166.74 /usr/bin/X -core 10 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp
2646 root      Ss  222M 1880 1.890 5 5.2 2.8 815.72 /opt/google/chrome/chrome
4519 root      Ss  1078M 244 1988 0 2.4 0.1 0.90 73.95cp
4181 root      Ss  1437M 327M 14480 10.2 4.5 0.43 0.7 /opt/google/chrome/chrome --type=renderer --lang-pt-BR --force-fieldtrials=
2757 root      Ss  407M 1540 15444 1.2 1.1 0.10 0.70 /opt/google/chrome/chrome --type=process --channel=2088.2.721943447 --en
3570 root      Ss  204M 114M 11488 0.8 1.1 2.27 56 /opt/google/chrome/chrome --type=renderer --lang-pt-BR --force-fieldtrials=
3583 root      Ss  593M 203M 16936 1.5 1.3 3.7 141.00 /opt/google/chrome/chrome --type=renderer --lang-pt-BR --force-fieldtrials=
2314 root      Ss  132M 117M 1612 1.1 1.4 0.90 80 compiz
4471 root      Ss  748M 196M 1616 1.5 1.3 1.1 0.00 31 /opt/google/chrome/chrome --type=renderer --lang-pt-BR --force-fieldtrials=
2644 root      Ss  122M 196M 1608 0.7 2.8 0.41 0.7 /opt/google/chrome/chrome
2744 root      Ss  407M 1540 15444 0.7 1.1 0.10 0.70 /opt/google/chrome/chrome --type=process --channel=2088.2.721943447 --en
4448 root      Ss  748M 196M 1616 1.5 1.3 0.02 58 /opt/google/chrome/chrome --type=renderer --lang-pt-BR --force-fieldtrials=
3818 root      Ss  621M 132M 1876 0.7 0.5 0.07 28 gnome-terminal
4447 root      Ss  748M 196M 1616 1.5 0.7 1.3 0.00 23 /opt/google/chrome/chrome --type=renderer --lang-pt-BR --force-fieldtrials=
1516 nobody    Ss  1028M 372 124 0 0.0 0.0 0.00 19 /usr/sbin/omniqa --no-resolv --keep-in-foreground --no-hosts --bind-Interfa
2837 root      Ss  1045M 402M 1052 3 0.9 6.5 0.00 58 /usr/sbin/psql
833 root      Ss  317M 1076 168 0 0.0 0.1 0.00 12 NetworkManager
712 root      Ss  415M 920 372 0 0.0 0.1 0.00 90 dbus-daemon --system --fork
2132 root      Ss  356M 154M 1090 0 0.0 0.2 0.10 37 /usr/bin/ibus-daemon --daemonize --xin
2227 root      Ss  451M 2613M 1088 0 0.4 0.07 50 /usr/lib/ibus/ibus-ut-gtki
1229 root      Ss  356M 154M 1090 0 0.0 0.1 0.10 60 /usr/bin/ibus-daemon --daemonize --xin
1 root      Ss  198M 456 700 0 0.0 0.1 0.01 99 /sbin/init
361 root      Ss  114M 184 8 0 0.0 0.0 0.00 30 ipset-udev-bridge --daemon
365 root      Ss  495M 812 856 0 0.0 0.1 0.00 40 /lib/systemd/systemd-udev --daemon
517 root      Ss  113M 168 132 0 0.0 0.0 0.00 13 upstart-socket-bridge --daemon
618 root      Ss  244M 132M 1452 0 0.0 0.2 0.00 25 smd
915 root      Ss  312M 156 132 0 0.0 0.1 0.00 60 /usr/sbin/ModemManager

```



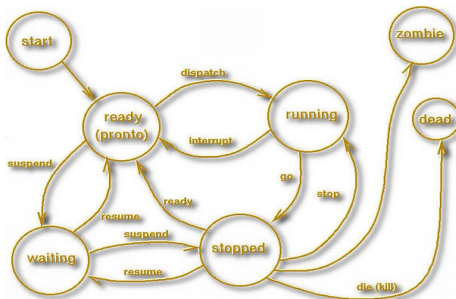
Um processo contém um contexto que o identifica:

- PID : um número inteiro que o indentifica
- UID : o usuário que o criou
- CPU : percentual da CPU que está utilizando
- MEM : percentual da memória que está sendo usado
- STS : o estado do processo naquele momento
- PRI : a prioridade que o processo tem para ser executado
- TMP : o horário que foi chamado
- CMD : o comando que o chamou
- xxx : ...

Estados de um processo

- Criado (ou iniciado)
- Apto (ou ready para ser executado)
- Executando (ou running)
- Esperando (ou waiting)
- Bloqueado (ou stopped)
- Destruído (ou dead)
- Zumbi (ou zombie)

Estados dos processos



O sistema operacional “pega” um processo na fila de aptos para receber o processador. O processo passa do estado “apto” para “executando”. O módulo do sistema operacional que faz essa seleção é o **Escalonador** (*Scheduler*)

Tipos de processos

Processos são criados e destruídos a todo tempo. O momento e a forma como eles são criados depende do sistema operacional considerado. A forma mais comum é permitir que os processos sejam criados livremente, através das chamadas de sistemas. A maioria de processos de um sistema operacional executam programas de usuários, entretanto alguns processos podem realizar tarefas do sistema.

Os processos do sistema (deamons) controlam autonomamente os recursos da máquina, são iniciados quando o computador ou quando chamamos, por exemplo, uma técnica chamada spooling copia os arquivos a serem impressos para a impressora, fazendo com que os arquivos na fila de impressão não precisam esperar pela impressora, dando a sensação de que o programa de usuário foi quem executou isso. Esse processo não está associado a nenhum programa de usuário, é próprio do sistema operacional.

Processos Interativos

São processos iniciados a partir de um sessão de terminal de comandos e por ele será terminado. Quando iniciamos um processo pelo terminal estamos rodando um processo em foreground. Um processo em foreground recebe diretamente do terminal(ou stdin) comandos que o controla. A saída(stdout ou stderr) também vêm via terminal. Um processo em background não estão apto a receber instruções diretamente, ele executa normalmente sem interrupções do usuário.

Processos em lote – *batch*

Processos em *batch* são controlados usualmente por um programa (cron no Linux). O programa Cron executa uma série de comandos pré-definidos usuário, exemplo: como o programa *i* at 18:30 xxx, será executado o programa xxx às 18h30. Uma série de comandos podem estar encadeados para serem executados em lote.

Simulação de um processo

Terminal de comandos do Linux com o comando **top** (ou htop)

Terminal de comandos do Linux com o comando **ps -aux**

Apagar as luzes, por favor !!!