Introdução à Linux

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Agenda

Agenda

- Introdução ao Linux
- Interação com o SO
- Gerenciando Processos
- Manipulando Arquivos
- Programando o SO

Sistema Operacional Linux

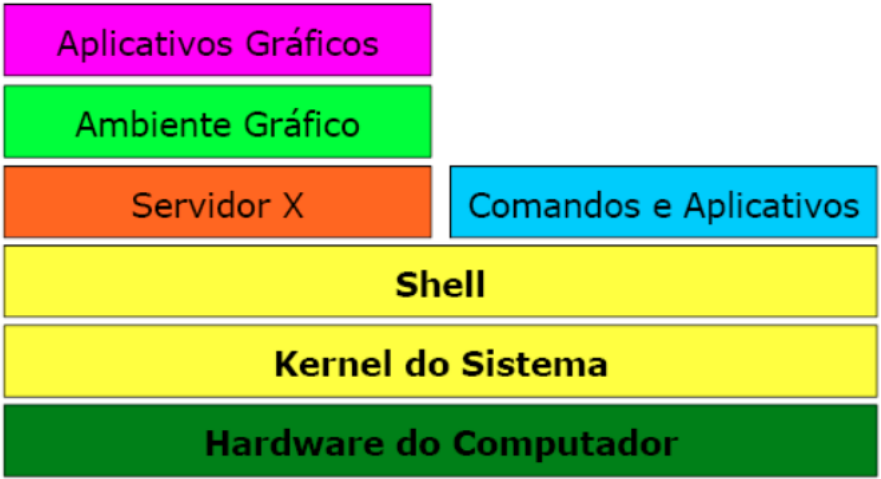
O Sistema Operacional (SO) é o programa que controla o computador, servindo de Interface entre o usuário e a máquina. O Sistema Operacional faz isso através de dois componentes: O Kernel e o Shell

- Kernel é o nome dado ao núcleo do Sistema Operacional. É o módulo deste programa que se comunica com o hardware do computador
- Shell é a "fachada" do Sistema Operacional. Essa é a parte do programa que se comunica com o usuário, recebendo seus comandos e repassando-os ao Kernel

Distribuição Linux

- É o nome dado ao conjunto de programas formado pelo Kernel Linux e por mais alguns softwares distintos (como Shells, aplicativos, jogos, utilitários, etc.)
- Várias empresas (ou pessoas) podem agrupar os programas que acham interessantes e criar suas próprias distros
- O Que Há Numa Distribuição?
  - Kernel, shell e ambiente gráfico
    - KDE, Gnome, ABlackBox, WindowMaker, Fluxbox

Sistema Operacional Linux



Sistema Operacional Linux

Sessão é um ponto de acesso ao shell do sistema.

- Iniciando uma sessão: ctrl+alt+F1 ou ctrl+alt+F2 ... ctrl+alt+F7
- Para encerrar uma sessão use o comando exit
- Toda configuração da sessão é perdida após ser encerrada, e vale apenas para a sessão corrente

Sistema Operacional Linux

- Ao iniciar uma sessão no linux o usuário tem acesso a um interpretador de comandos que chamamos de bash
- Se a sessão foi iniciada na interface gráfica, é necessário abrir o terminal para ter acesso ao bash
- O bash pode ser programado e controlado por meio de variáveis de ambiente e pelo uso de scripts que coordenam a execução de diversos comandos

Sistema Operacional Linux

Comandos de sessão:

- Login: iniciar sessão: Interface Gráfica ou Terminal
- Logoff : sair da sessão
- Reboot : reinicia o sistema
- poweroff ou shutdown –h now desliga o sistema

Atalhos úteis em uma sessão:

- TAB auto completa comandos
- History mostra a lista de comandos executados
- ! id-histórico executa o comando do histórico com o id id-histórico
- Man mostra ajuda de comandos

Variáveis de Ambiente para parametrizar uma sessão:

- export : cria uma variável
- env : mostra as variáveis criadas
- unset : apaga uma variável
- echo : mostra o valor atribuído a uma variável

Gerenciando Processos

Tipos de processos no Linux:

- Processos interativos: são iniciados a partir de uma sessão de usuário no terminal de comandos e são controlados por ele
- Processos em lote (batch): o processo entra em execução e não permite nenhuma interação com o usuário do SO
- Daemons: são processos servidores normalmente executados quando o Linux é inicializado, permanecendo em execução enquanto o sistema estiver em funcionamento esperando em background que outro processo solicite o seu serviço.
  - Utiliza-se o operador & para que o processo execute e libere o terminal

Gerenciando Processos

Pesquisando processos em execução comando ps

- Exibe informações sobre os processos ativos
- ps [opções]
- -a exibe também informações de outros usuários
- -u exibe o nome do usuário e a hora de início do processo
- -x exibe também os processos não associados a um terminal de controle
- -p pid exibe o processo cujo número é pid

Gerenciando Processos

kill: finaliza um processo por meio do pid;

- kill [opções] [sinal] pid
- -n sinal aplicado ao processo
- -l lista todos os nomes e números de sinais
- ps -aux
- kill -9 1029

Gerenciando Processos

Comandos para exibir informações sobre o computador de forma geral

- free: exibe a quantidade de memória livre;
- lscpu: exibe informações sobre o CPU

Manipulando Arquivos

Estrutura de arquivos

- Referência a um arquivo/diretório é feita através de um caminho, que é formado da seguinte forma:
- [diretório raiz] [diretório 1] [diretório 2] ... [diretório n] [arquivo]
- Cada um desses itens são separados por uma / (barra) no Linux
- o diretório raiz chama-se / (barra)
- exemplo de caminho: /home/silvio/doc/1

Manipulando Arquivos

Alguns diretórios possuem notações especiais ou "atalho":

- ~ til : ao referir-se ao diretório ~ (til), o sistema entende como o diretório pessoal do usuário, ou seja, /home/[usuário], onde [usuário] é o nome de login do usuário atual.
- -: o kernel Linux armazena um histórico dos diretórios que acessamos. O - (hífen) refere-se ao último diretório acessado
- .: o símbolo . (ponto) refere-se ao diretório atual, ou seja, aquele em que estamos trabalhando
- ..: o .. (ponto ponto) refere-se ao diretório acima do qual estamos.

Manipulando Arquivos

No Linux existem três tipos de permissão para definir os acessos a arquivos e diretórios:

- r : Permissão de leitura para um arquivo; e permitir listar o conteúdo de um diretório através do comando ls diretório
- w : Permissão de gravação e exclusão para um arquivo/diretório.
- x : Permissão para executar um arquivo, se for um arquivo binário ou um script; e se for um diretório permitir acesso a ele através do comando cd diretório
- Tais permissões são concedidas aos seguintes papéis de usuário
  - Dono : O proprietário do arquivo ou criador do arquivo
  - Grupo: Usuários que fazem parte do grupo do proprietário
  - Outros: Não são os proprietários e nem fazem parte do grupo

Manipulando Arquivos

Comandos para Manipulação de Arquivos

- cd : Comando para acessar um diretório
- pwd : retorna o caminho do diretório desde a raiz /
- cp : copia um arquivo ou diretório de uma caminho para outro
- find : procurar por arquivos ou diretórios no sistema de arquivos
- mkdir : cria um novo diretório
- mv : mode um arquivo ou diretório de um caminho para outro
- ls : lista um diretório

Comandos para Manipulação de Arquivos

- rm : remove um diretório, desde que esteja vazio
- touch : cria um arquivo vazio
- du : retorna o espaço utilizado por um diretório ou arquivo
- df : retorna as partições presentes no sistema
- tree : retorna a árvore do sistema
- chmod : muda a permissão de arquivos e diretórios

Manipulando Arquivos

Fluxos de Entrada e Saida

- Stdin (0) – Entrada Padrão
- Stdout (1) – Saída Padrão
- Stderr (2) – Saída de Erro
- Operadores
  - – > < >> << ; |

Manipulando Arquivos

cat : concatena arquivos e lista na saida padrão Abreviação para concatenate

- Sintaxe: cat [opções] parametros
- Parâmetros pode ser uma lista contendo arquivos
- Exemplo de uso: cat arq1.txt arq2.txt
  - Concatena arq1.txt e arq2.txt e exibe na saída padrão
- Outro Exemplo: cat > arq3.txt

Manipulando Arquivos

wc: conta palavras, linhas, caracteres de um arquivo texto

- Sintaxe: wc [opções] arquivo
- Exemplo: dmesg — wc –l
- Exibe a quantas linhas (-l) existem no log do kernel (dmesg)
  - Principais opções:
  - -l Lines : exibe o número de linhas do arquivo
  - -w Words : exibe o número de palavras

Manipulando Arquivos

head: imprime as primeiras linha de um arquivo

- Sintaxe: head [opções] arquivo
- Exemplo de uso: dmesg — head 3
- Exibe as três primeiras (-3) linhas do log do kernel (dmesg)
- Principais opções:
  - -n lines N : exibe as primeiras N linhas do arquivo

Manipulando Arquivos

tail: Imprime as ultimas linhas de um arquivo

- Sintaxe: tail [opções] arquivo
- Exemplo de uso: dmesg — tail 3
- Exibe as três ultimas (-3) linhas do log do kernel (dmesg)
- Principais opcões:
  - -n lines N : exibe as últimas N linhas do arquivo

Manipulando Arquivos

sort: ordena as linhas de arquivo texto

- Sintaxe: sort [opções] arquivo
- Outro uso comum: comando — sort
- Principais opções:
  - f ignorecase: campos que serão exibidos
  - r reverse: ordena na ordem inversa

Manipulando Arquivos

uniq: reporta ou omite linhas repetidas

- Sintaxe: uniq [opções] arquivo
- Outro uso comum: comando — uniq
- Principais opções:
  - c count: informa a quantidade de ocorrências
  - i Ignorecase: campos que serão exibidos
  - d Repeated : exibe só o que tiver repetição
  - u Unique : exibe só que tiver uma única ocorrencia

Programando o SO

O Bash permite a execução de scripts

- Para isso basta criar um arquivo texto, e executar com o interpretador bash
- Exemplo: /bin/bash script.sh
- Se alterar o modo do arquivo script.sh para executável, é possível executa-lo diretamente:
  - ./script.sh

Programando o SO

Programação em Bash

- Função
- Condicional
- Loop

Desenvolvimento Contínuo

Funções são blocos de comandos definidos para serem chamados em outras partes do código.

Sintaxe:

```
1
2  nome_da_funcao() {
3    comandos
4  }
5
6  nome_da_funcao
```

Exemplo de função

```
1
2  funcao1() {
3    ls /tmp
4    touch /tmp/out
5    echo date > /tmp/out
6  }
7
8  funcao1
```

If definição de condição binária

Sintaxe:

```
1
2  if [ condicao ] then
3    bloco 1
4  else
5    bloco 2
6  fi
```

Exemplo de if

```
1
2 if [ -e $linux ] then
3   echo 'A variavel $linux existe.'
4 else
5   echo 'A variavel $linux nao existe.'
6 fi
```

Tabela de opções de condicional

-eq	Igual
-ne	Diferente
-gt	Maior
-lt	Menor
-o	Ou
-d	Se for um diretório
-e	Se existir
-z	Se estiver vazio
-f	Se conter texto
-o	Se o usuário for o dono
-r	Se o arquivo pode ser lido
-w	Se o arquivo pode ser alterado
-x	Se o arquivo pode ser executado

Desenvolvimento Contínuo

for é um comando que executa um laço contado. A quantidade de iterações e valor de cada iteração é definida pelo comando passado após a palavra in

```
1
2 for variavel in comandos do
3   comando
4 done
```

Desenvolvimento Contínuo

Exemplo de Loop

```
1
2 for i in * do
3   cp $i /tmp
4 done
```