0Atividades Semb2

Semana 1

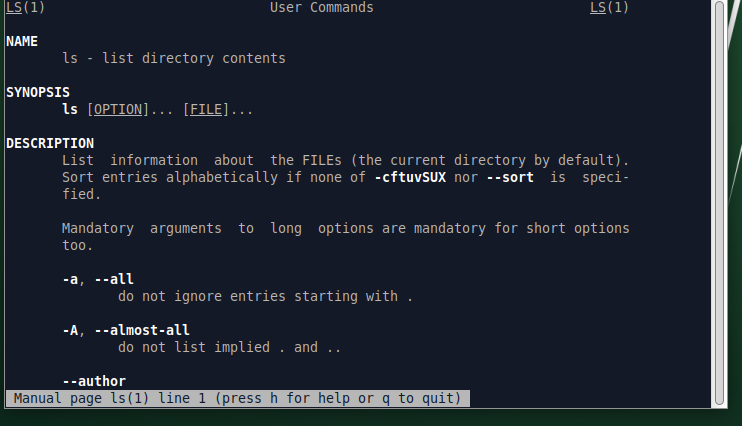
2. Listagem de comandos do terminal Linux

**a)**

**ls:** É um comando de listagem usado para mostrar as informações dentro de uma pasta.



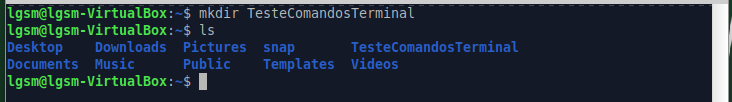
**man:** Utilizado para descrever a funcionalidade do programa passado, como se fosse para ver um manual do comando.



**Clear:** limpa o terminal, pode ser usada pelo atalho Ctrl + l.



**mkdir:** Cria um novo diretório com o nome passado (mkdir nomeNovoDiretorio1 nomeNovoDiretorio2 “nome Novo Diretorio3”).



**cd:** Permite a transição entre diretórios (cd DiretorioDesejado).



**pwd:** Imprime completamente o diretório atual de trabalho.



**whoami:** Imprime o usuário em uso no terminal no momento do uso do código.



**Redirecionadores:** Usando-se >> é possível pegar a informação de um comando escrito anteriormente a ele e passar essa informação para outro arquivo, o qual se não existir também será criado.



**Criar e acessar pasta com nome composto e exemplo:** Para se criar uma pasta com nome composto pode ser usado mkdir “Nome da pasta Composto”. Para se navegar para esta passa, será usado a contrabarra(\) antes do espaço para este ser reconhecido, como em cd Nome\ da\ pasta\ Composto/. Também pode ser digitado o nome da pasta entre aspas (cd “Nome da pasta Composto”).

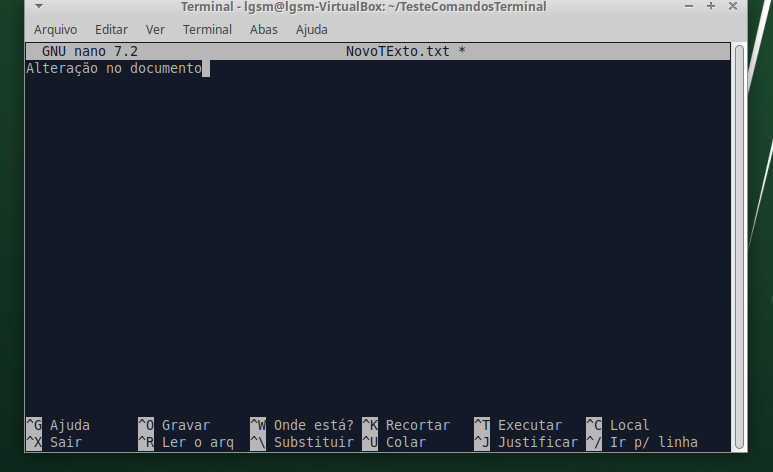


**touch:** Atualiza os horários de modificação de acesso de cada arquivo passado para o horário atual. Caso não exista o arquivo passado como argumento ele será criado vazio.



**nano:** Abre o editor de texto nano no terminal, para alteração do arquivo passado como argumento.

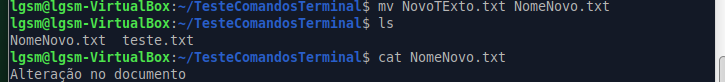




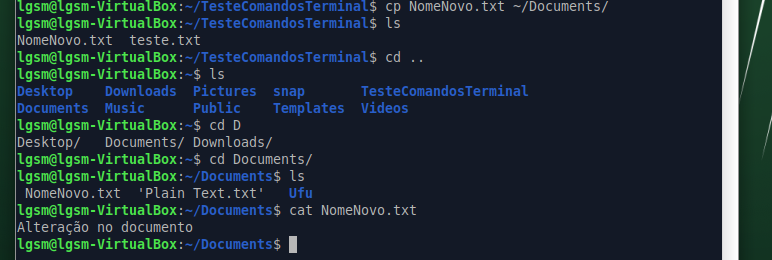
**cat:** Concatena arquivos na saída. Pode ser usado para se obter a informação de um arquivo e imprimi-la no terminal.



**mv:** move o arquivo passado para o destino passado, o qual pode ser em outro diretório. Este comando também pode ser usado para renomear arquivos.



**cp:** Copia o arquivo passado no diretório escrito (cp ArquivoParaCopiar Diretorio).



**find:** Encontra o arquivo ou diretório passado como argumento, a partir do diretório em que foi usado o comando (find . -name nomeDoAquivoParaProcura).



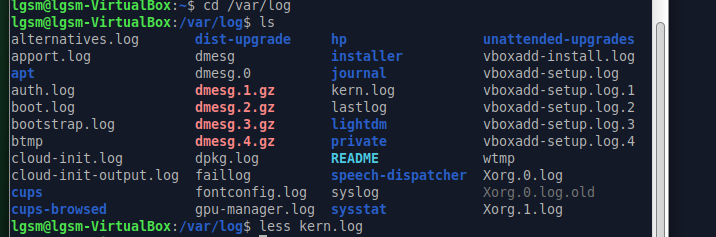
**head:** Apresenta o cabeçalho (primeiras linhas) do arquivo de texto passado como argumento.

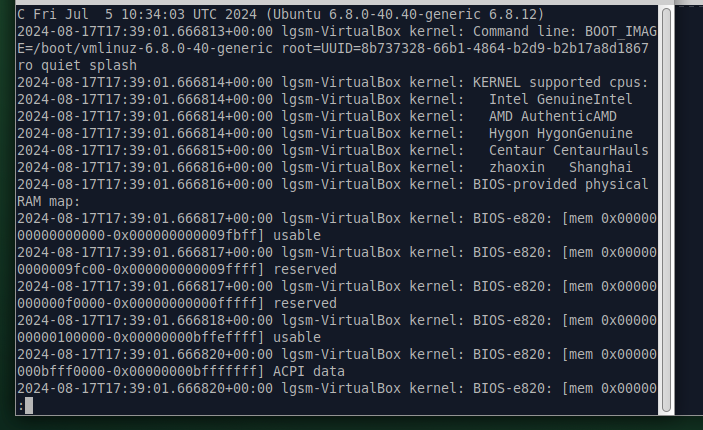


**tail:** Apresenta as últimas linhas do arquivo de texto passado como argumento.



**less:** Permite a leitura de um arquivo por inteiro, mas carregando este por partes. Isso é útil para que o computador não tente ler um arquivo grande por inteiro que consumiria muito dele.

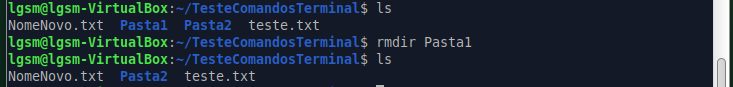




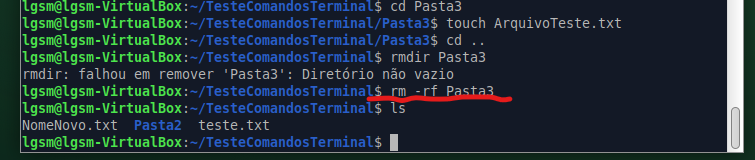
**rm:** Remove o arquivo passado como argumento, mas não apaga diretórios por padrão.



**rmdir:** Remove o diretório passado como argumento, por padrão remove apenas diretórios vazios.



**rm -rf:** Ao usar o parâmetro -rf o rm apagará o argumento passado por inteiro, mesmo se este estiver com outros arquivos.



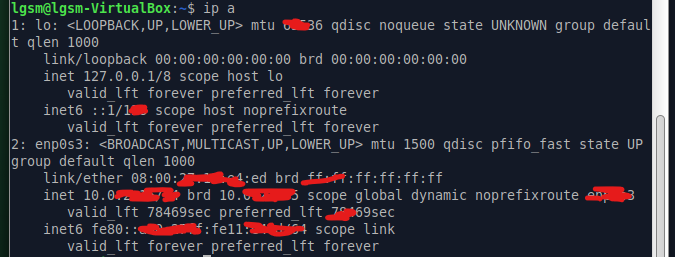
**hostname:** Imprime o nome da máquina.



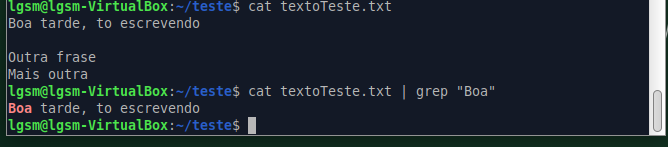
**hostname -I:** Imprime todas as informações de IP da máquina.

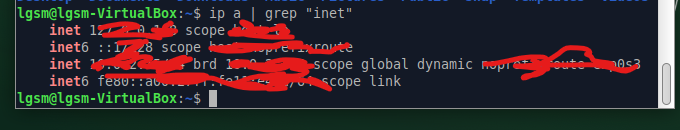
****

**ip a:** Mostra todos os detalhes de todos os endereços ip da máquina.

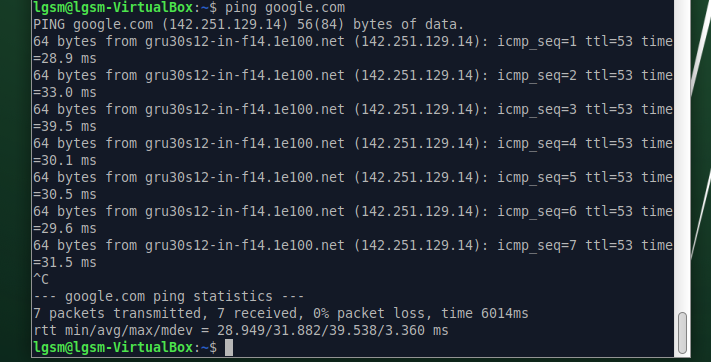
****

**grep:** Este código busca por padrões em arquivos. Desse modo, ele pode ser usado para filtrar a busca por textos específicos em arquivos. Para esse uso, deve ser escrito | (piping) anteriormente ao grep, pois assim será pega a saída de um outro programa para o grep.



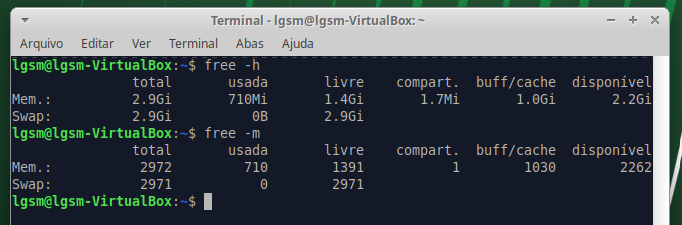


**ping:** Usado para observar se está tendo comunicação pelo servidor (ip) passado (ping IpDoServidor).

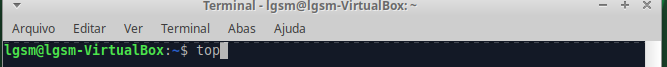


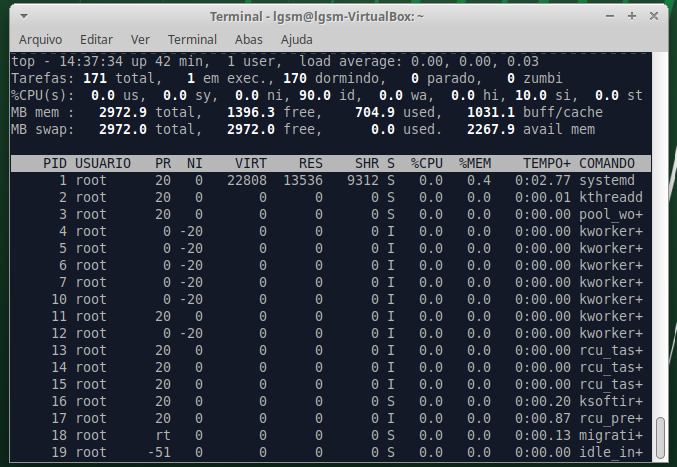
**Contrl + c:** Este atalho permite que qualquer comando em execução no terminal seja cancelado.

**free -h e free -m:** Esses comandos são usados para uma leitura rápida e resumida do uso da memória pelo sistema. O parâmetro -h mostra os dados de memória de um modo mais humano de leitura e o -m mostra esses parâmetros em megabites.



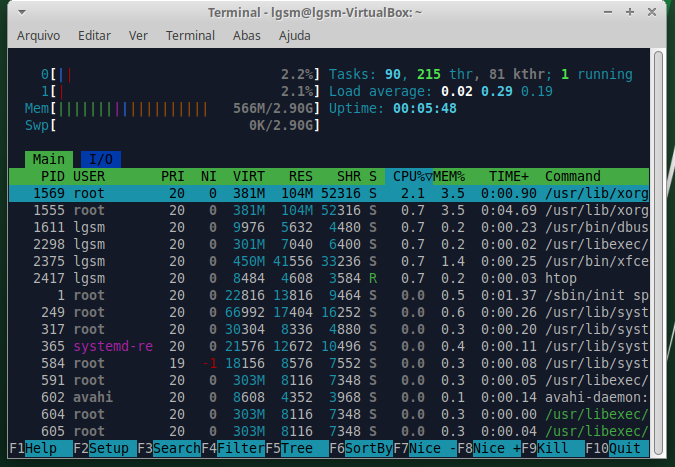
**top:** [O comando top é uma forma de ver os processos em execução no Linux, com atualização de tela e ordenado pelo processo que utilizam mais CPU](https://www.bing.com/ck/a?!&&p=5472d1e6a26873ffJmltdHM9MTcyNDgwMzIwMCZpZ3VpZD0xNzFmNDNkYy05YjY4LTY5YjItMjlkZC01NzAyOWFhNjY4ODcmaW5zaWQ9NTcyOA&ptn=3&ver=2&hsh=3&fclid=171f43dc-9b68-69b2-29dd-57029aa66887&psq=comando+top+linux&u=a1aHR0cHM6Ly93d3cuY2VydGlmaWNhY2FvbGludXguY29tLmJyL2NvbWFuZG8tbGludXgtdG9wLw&ntb=1).

****

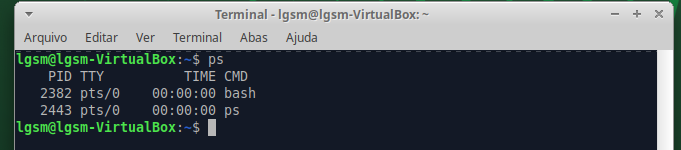
****

**htop:** É um comando usado para que permite o monitoramento dos sistemas no indux em tempo real. Ele apresenta informação de uso de memória, núcleos de processadores e permite a filtragem e busca de determinados processos desejados. (Basicamente um gerenciador de tarefas).

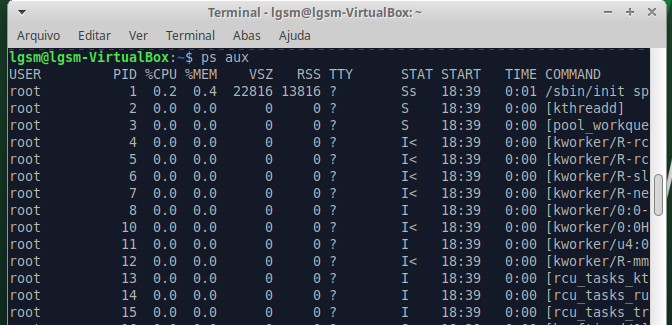


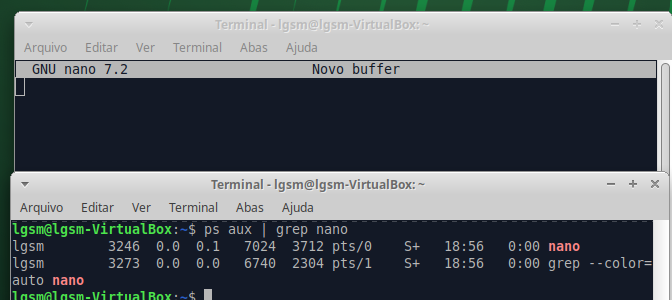


**ps:** Este comando mostra os processos rodando na instância do terminal em que ele foi utilizado.

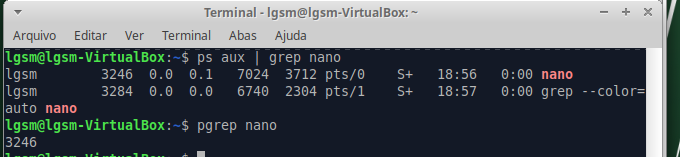


**ps aux:** Apresenta todos os processos rodando na máquina no instante que o comando foi utilizado. Ele pode ser usado junto com o comando grep para filtragem de algum processo específico. (ps aux | grep processoDesejadoParaFiltragem)

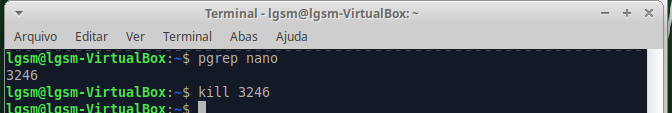




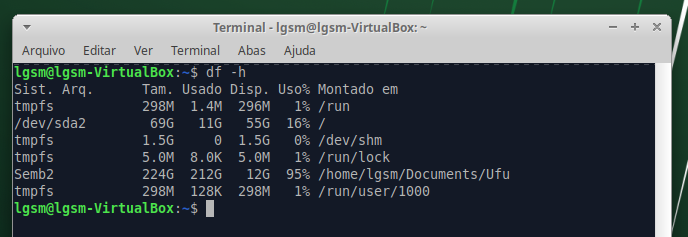
**pgrep:** Este comando retorna o pid (id de processo) do processo passado como argumento se este estiver em execução.



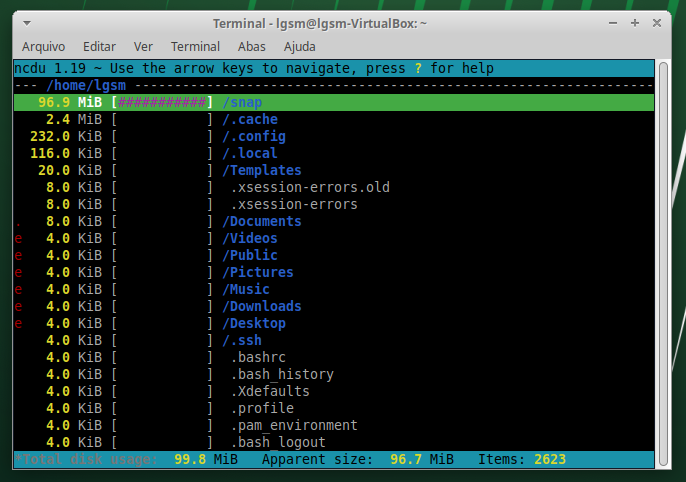
**kill:** Comando que finaliza um processo. Ele deve receber como argumento o pid do processo desejado para encerramento.



**df -h:** Ele fornece informações sobre o espaço de armazenamento disponível e utilizado em todas as partições e sistemas de arquivos montados no sistema.

****

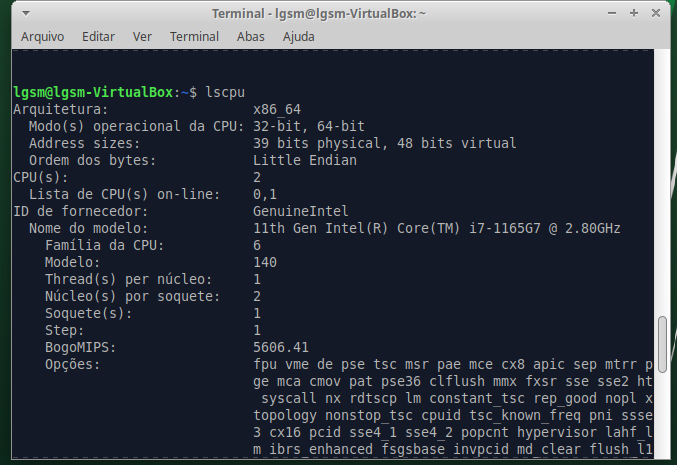
**ncdu:** Ele serve para análise do uso de espaça do disco de armazenamento analisando e rastreando quais arquivos e diretórios estão usando seu espaço em disco no Linux.

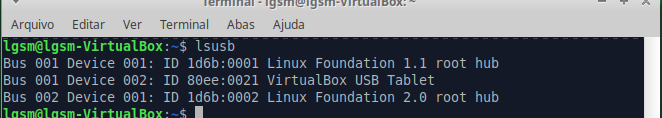


**uname:** apresenta informações sobre o kernel do sistema.

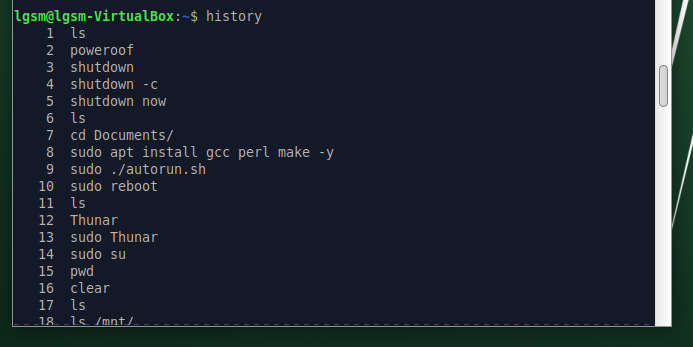


**lscpu e lsusb:** O primeiro mostra informações a respeito do processador da máquina e o ultimo informações a respeito das conexões usb da máquina.



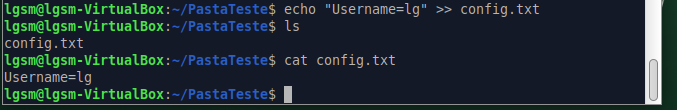


**history:** Apresenta um histórico de todos os comandos utilizados no terminal.



**b)**

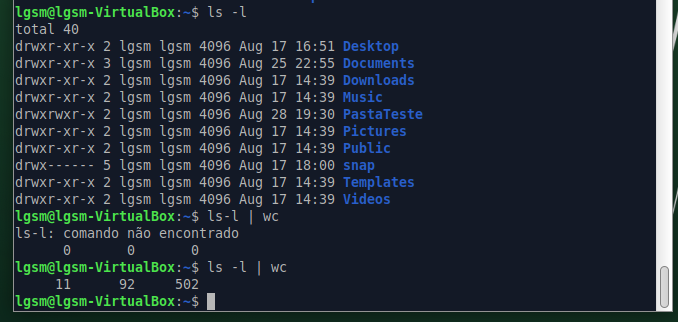
**echo:** Este comando mostra imprime (ecoa) o argumento passado na saída padrão seguido por uma nova linha. Ele pode ser usado para se iniciar um arquivo com algum conteúdo já escrito.



**wc:** Conta o número de linhas, palavras e bytes (apresentando-os nesta ordem) do argumento passado.

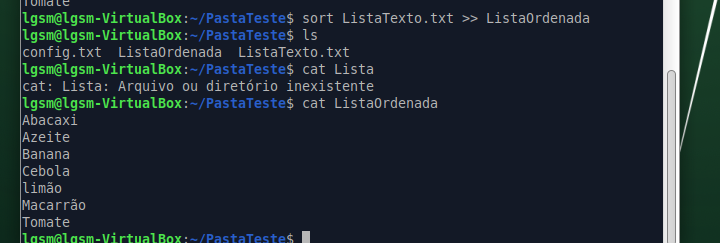


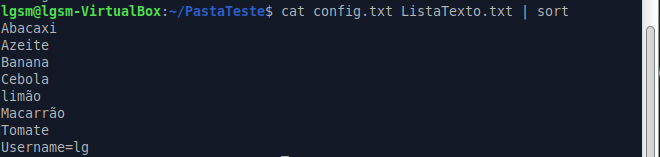
**piping:** Pega a saída de um comando e o passa como a entrada de um outro comando.



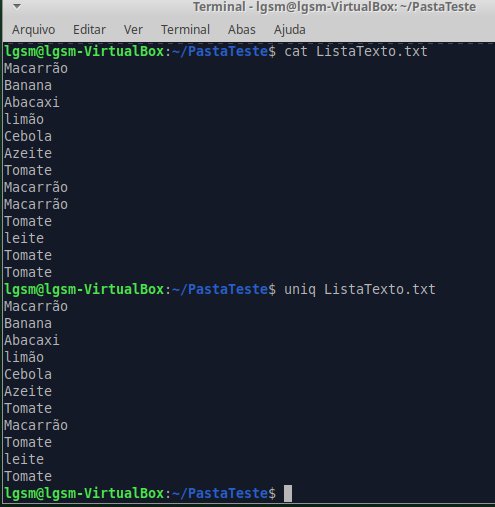
**sort:** Este comando imprime as informações de um arquivo de maneira ordenada, mas não altera o arquivo em si.

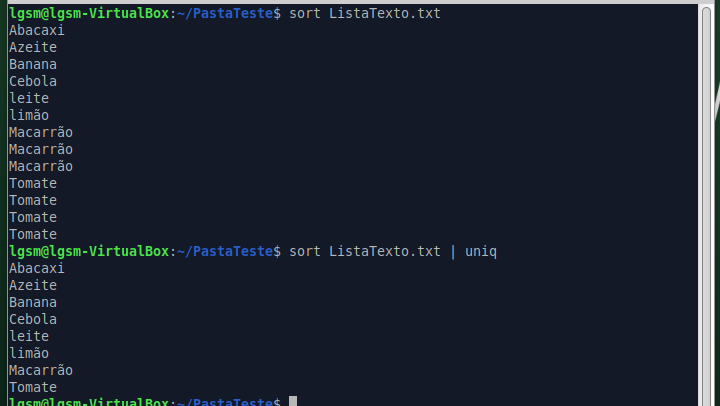




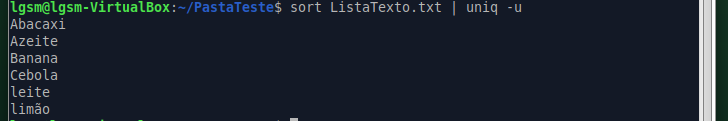


**uniq:** Ele reporta (se passado o parâmetro -d) ou omite linhas adjacentes repetidas. Pode ser usado junto do comando sort para com isso imprimir os valores de uma linha apenas uma vez. Pode ser usado com o parâmetro -u para imprimir os valores que não se repetem.

****

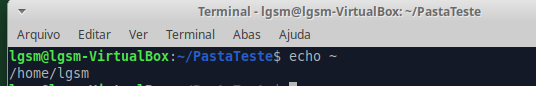
****

****

****

**expansions:** Alguns caracteres possuem significados especiais os quais podem ser expandidos se usados no terminal.

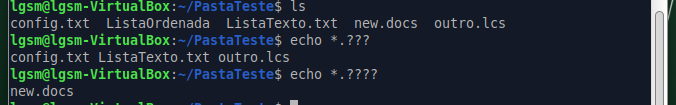
O caractere ~ apresenta a informação da diretório home do usuário.



O caractere \* retorna todos os nomes de caminhos e arquivos presentes no diretório em que foi utilizado.



O caractere ? representa que deve-se ter um caractere qualquer na posição em que foi posto o ?.



A expansão {} de chaves é um mecanismo para gerar strings a partir de padrões.

