



Comandos do Linux

Métodos de Programação II

MIETI :: 2015/16

António Esteves

DI/EEUM/UM



Comandos do Linux

■ Obter ajuda:

- `man comando` → mostra o manual relativo ao comando especificado; para terminar a visualização deve-se premir 'q'

■ Mudar de pasta/diretório:

- `pwd` → mostra o caminho da pasta atual.
- `cd nomeDir` → mudar para a pasta especificada.
 - “.” → pasta atual
 - “..” → pasta 'pai' da pasta atual.



Comandos do Linux

■ Ver o conteúdo de ficheiros e pastas:

- `ls nomeDir` → lista os ficheiros da pasta.
- `ls -al nomeDir` → lista os ficheiros da pasta, mostrando todos os atributos dos ficheiros.
- `cat nomeFich` → mostra o conteúdo do ficheiro.
- `more nomeFich` → mostra o conteúdo do ficheiro, com paragem ao fim de cada página (ecrã) mostrada.



Comandos do Linux

■ Controlo sobre ficheiros e pastas:

- **cp** fichOrig fichDest → copia um ficheiro para outro
- **mv** fichOrig fichDest → move o ficheiro ou muda-lhe o nome
- **rm** nomeFich → elimina o ficheiro
- **rm -r** nomeDir → elimina a pasta e todas as suas sub-pastas
- **mkdir** nomeDir → cria uma pasta
- **rmdir** nomeDir → elimina uma pasta vazia
- **ln** opcoes fichAlvo nomeLink → cria um atalho para um ficheiro
- **chmod** → altera as permissões do utilizador, do grupo, outros.
RWX ⇔ R=ler (peso 4), W=escrever (peso 2), X=executar (peso 1)

Exemplo: **chmod 764** nomeFich → o utilizador pode RWX o ficheiro, o grupo pode RW o ficheiro e os outros podem R ficheiro.



Comandos do Linux

■ Pesquisa:

- `find` → procura recursiva de ficheiros
- `grep` → procura um dado padrão num ficheiro

■ Editores:

- `vi` `fich``Editar` → editor de ficheiros orientado-ecrã



Comandos do Linux

■ Terminal/consola:

- **history** → mostra a lista de comandos executados
- **clear** → limpa o ecrã
- **exit** → termina um processo; se não houver processos em execução, sai da consola.

■ Compressão:

- **tar** → compacta vários ficheiros num único
- **gzip** → utilitário de compressão da GNU
- **bzip2** → utilitário de compactação e compressão
- **unzip** → descomprime ficheiros zip



Comandos do Linux

- `bzip2 -d -v nome_ficheiro.bz2` → descomprimir um ficheiro tipo BZ2
- `bzip2 -zvf nome_ficheiro_original` → comprimir um ficheiro em formato BZ2
- `gzip -dvf nome_ficheiro.tar.gz` → descomprimir um ficheiro tipo GZ
- `gzip -vf nome_ficheiro_original` → comprimir o ficheiro 'nome_ficheiro_original' num ficheiro tipo GZ com nome 'nome_ficheiro_original.gz'
- `tar -xvf nome_ficheiro.tar` → descompactar um ficheiro tipo TAR
- `tar -cvf novo_ficheiro.tar file1 file2 ... fileN` → compactar os ficheiros file1, file2, ..., fileN num ficheiro tipo TAR
- `tar -cvf novo_ficheiro.tar dir_files` → compactar o conteúdo da pasta 'dir_files' num ficheiro tipo TAR



Comandos do Linux

■ Informação sobre o utilizador:

- `passwd` → muda a palavra passe do utilizador
- `who` → mostra quem está ligado no sistema

Exemplo: '`who am i`'

■ Utilização do sistema:

- `ps` , `top` → mostra os processos a correr
- `kill processID` → mata o processo com o ID indicado
- `uptime` → indica há quanto tempo o sistema está a funcionar



GCC: GNU C/C++ Compiler

- Manual online: gcc.gnu.org/onlinedocs/gcc/index.html
- Manual no Linux/Cygwin: `man gcc`
- Exemplo:
`gcc -g -O0 -Wall inFile.c -o outFile.o`
- `-g` → gera informação de depuração. Por exemplo, o GDB consegue utilizar esta informação.
- `-Wall` → ativa a emissão da maior parte dos avisos por parte do compilador
- `-O0` → sem otimização (também existe `-O1`, `-O2`, `-O3`)
- `-o outFile.o` → especifica o ficheiro de saída.
- Mais opções do compilador:
 - incluir vários ficheiros fonte (.c)
 - especificar pastas a usar na pesquisa dos ficheiros
 - controlar o tipo de saída da compilação (*assembly*, objeto, executável)
 - controlo da informação de depuração
 - controlar a otimização
 - controlar o *linking*
 - relativas à arquitetura alvo (ARM, i386, x86-64, IA-64, MIPS, ...)
 - etc.



GDB: *GNU Debugger*

Alguns comandos:

- **break numeroLinha** → cria um ponto de paragem na linha especificada
- **break nomeFicheiro:numeroLinha** → cria um ponto de paragem na linha do ficheiro especificados
- **run** → executa o programa
- **c** → continua a execução
- **next** → executa a próxima linha
- **step** → executa a próxima linha ou entra dentro da função
- **quit** → sai do **gdb**
- **print expressao** → escreve no ecrã o valor atual da expressão especificada
- **help comando** → ajuda sobre o comando especificado



Comandos do Linux

■ Mais sobre os comandos do Linux:

- www.matchstick.com/unix/core.html
- www.nmr.chem.uu.nl/~abonvin/tutorials/MD-Data/unix/
- usar o comando man.