Linux: Comandos

Sistemas de Computação

LCC :: 2019/20

António Esteves

DI/UM

Comandos mais utilizados do Linux

- Obter ajuda:
  - man comando → mostra o manual do comando especificado (premir '**q**' para sair da visualização do manual)

- Mudar de pasta:
  - pwd → mostra o caminho da pasta atual;
  - cd nomeDir → mudar para a pasta indicada.
    "." → pasta atual;
    ".." → pasta acima em direção à raiz ("/").

Comandos mais utilizados do Linux

- Visualizar o conteúdo de ficheiros/pastas:
  - ls *nomeDir* → mostrar a lista de ficheiros da pasta

    ls -al nomeDir → mostrar (com todos os atributos) a lista de ficheiros da pasta
  - cat nomeFicheiro → mostrar o conteúdo do ficheiro
  - more nomeFicheiro → mostrar o conteúdo do ficheiro, parando cada vez que preencher o ecrã na totalidade

Comandos mais utilizados do Linux

- Operações sobre ficheiros/pastas:
  - cp fichOrig fichDest → copiar um ficheiro
  - mv fichOrig fichDest → mover/mudar nome dum ficheiro
  - rm nomeFich → eliminar um ficheiro

    rm -r dirNome → eliminar uma pasta e as subpastas
  - mkdir dirNome → criar uma pasta
  - rmdir dirNome → eliminar uma pasta vazia
  - ln opcoes alvo nomeAtalho → criar um atalho
  - chmod → alterar as permissões do utilizador/grupo/outros. RWX ⇔ **R**-ler (peso 4), **W**-escrever (2), **X**-executar (1).
  - Exemplo: chmod 764 nomeFich → o utilizador pode RWX ficheiro, o grupo pode RW ficheiro, e os outros podem R ficheiro.

Comandos mais utilizados do Linux

- Pesquisa:
  - find → procura recursiva de ficheiros
    Exemplo: find */home -name tpc1.pdf*
  - grep → procura um dado padrão num ficheiro
- Editores:
  - nano fichEditar → editor de ficheiros orientado ao ecrã
  - vi, vim

Comandos mais utilizados do Linux

- Sessão e consola:
  - history → mostrar os comandos utilizados
  - clear → limpar o ecrã
  - exit → terminra a sessão atual. Se não tivermos nenhum processo a correr, sai da consola/shell.

- Compressão:
  - tar → arquivar/juntar vários ficheiros num único
  - gzip → utilitário de compressão da GNU
  - bzip2 → outro utilitário de compressão
  - unzip → descompactar ficheiros zip

Comandos mais utilizados do Linux

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

Comandos mais utilizados do Linux

- Informação sobre o utilizador:
  - passwd → muda a palavra passe do utilizador
  - who ou w → 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 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 inFileName.c -o outFileName.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 outFileName.o → especifica o ficheiro de saída.
- Mais opções do compilador:
  - incluir vários ficheiros fonte (.c)
  - especificar diretorias 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

Linux: Comandos

- Mais sobre comandos do Linux:
  - www.matchstick.com/unix/core.html
  - www.nmr.chem.uu.nl/~abonvin/tutorials/MD-Data/unix/
  - usar o comando man.