

常用命令速查表

Linux系统

- `ls`：用于列出当前目录下的所有文件，`ls -l` 可以显示详细信息
- `pwd`：能够列出当前所在的目录
- `cd DIR`：可以切换到 `DIR` 目录，在Linux中, 每个目录中都至少包含两个目录，`.` 指向该目录自身，`..` 指向它的上级目录，文件系统的根是 `/`
- `touch NEWFILE`：可以创建一个内容为空的新文件 `NEWFILE`，若 `NEWFILE` 已存在，其内容不会丢失
- `cp SOURCE DEST`：可以将 `SOURCE` 文件复制为 `DEST` 文件；如果 `DEST` 是一个目录，则将 `SOURCE` 文件复制到该目录下
- `mv SOURCE DEST`：可以将 `SOURCE` 文件重命名为 `DEST` 文件；如果 `DEST` 是一个目录，则将 `SOURCE` 文件移动到该目录下
- `mkdir DIR`：能够创建一个 `DIR` 目录
- `rm FILE`：能够删除 `FILE` 文件；如果使用 `r` 选项则可以递归删除一个目录。删除后的文件无法恢复，使用时请谨慎
- `man`：可以查看命令的帮助 例如 `man ls` 可以查看 `ls` 命令的使用方法；灵活运用 `man` 和互联网搜索，可以快速学习新的命令
- `./程序名`：运行可执行文件。如果你在 `x86_64` 的 Linux 操作系统上通过 `gcc hello.c` 生成了一个 `elf` 格式的可执行文件 `a.out`，你可以通过 `./a.out` 运行这个程序

推荐[Linux入门教程](#)。

TMUX 终端复用器

- `tmux` 打开 `tmux` 模式
- `tmux` 模式下操作
 - `Ctrl+b` 一切操作的先导键，之后的命令被识别成 `tmux` 的默认命令 下面的所有命令都是以这个快捷键开头
 - `Shift+5 = %` 将当前会话横向分割成为两个
 - `Shift+' = "` 将当前会话纵向分割成为两个
 - `Shift+o = o` 切换到下一个会话
 - `Shift+x = x` 删除当前会话
 - `Shift+w = w` 查看所有会话

更多操作请参见 [cheatsheet](#) 或 [STFW](#)。

GDB 调试器

Linux 平台下最常用的一款程序调试器。本说明仅简单介绍一些常用的指令，其他操作可以STFM或者查看[相关文档](#)。

对于 `print` 大法而言，其在相对较多情况下是极为有用的，在细粒度的调试上有所不足，比如说当编译器出现段错误的时候，此时使用 `gdb` 来进行调试，并且切换栈帧，找到出错的地方相对比较方便。

- 配置 `gdb` 相关：
 - `file`: 使用filename作为要调试的程序。它是为了它的符号和纯粹记忆的内容而阅读的。
 - `symbol-file`: 从文件filename读取符号表信息。必要时搜索PATH。使用 `file` 命令从同一个文件中获取符号表和要运行的程序。
 - `add-symbol-file`: 从文件文件名中读取附加符号表信息。当filename被动态加载（通过其他方式）到正在运行的程序中时，可以使用此命令。
 - `target remote :26000`: 连接本地端口26000的qemu。
- 程序运行：
 - `b: breakpoint: b location [if condition]` 12
 - `location`: filename:line_num, +/- off-set, function-name, file_name-function_name
 - `condition`: bool表达式，计算为真的时候中断
 - `d: delete`: 删除断点
 - `info`: 显示一些有关于正在调试程序的一般指令
 - `info mem`: 显示内存区域属性
 - `info reg`: 显示部分寄存器
 - `info all-registers`: 显示全部寄存器
 - `info breakpoint`: 显示当前的所有断点
 - `watchpoint: watch [-l|-location] expr [thread thread-id] [mask maskvalue] [task task-id]`
 - `bt`: 查看当前调用栈
 - `frame number`: 切换当前调用栈
 - `layout name`: 更改当前 TUI 窗口显示的内容。{next, prev, src, asm, split, regs}
 - `c: continue`
 - `s: step`: 单步执行, 进入下一行的函数调用中
 - `n: next`: 单步执行, 不进入函数调用
 - `p: print expr`

- `Ctrl+C`: 暂停执行
- `Ctrl+D`: 退出gdb

推荐的帮助说明有[cheat sheet](#) 和 [Debugging with GDB](#) 还有 [gdb常用指令](#)。

QEMU 模拟器

纯软件实现的虚拟化模拟器，几乎可以模拟任何硬件设备，但是由于其基于软件实现，而非硬件，因此效率相对较低。

- `Ctrl+A x` 退出 qemu
- `Ctrl+A c` 打开 `qemu monitor console`，可以执行一些命令查看当前模拟机器的运行状态，控制虚拟机的各个方面。
 - `info mem` 列出活动虚拟内存映射
 - `info registers` 列出CPU寄存器

就本课程的内容来说，其提供的支持实际上不如 `gdb` 提供的方便，更多信息可见[文档](#)。