# Die wichtigsten Befehle für die Konsole, Vim und die (SGE) Queue

## Konsole

| | |
|---|---|
| man Befehl | Zeit Hilfe für den Befehl an, Navigation über Pfeil-Tasten oder mit *j*/*k*, suchen mit */...*, beenden mit *q* |
| ls [-alh] | zeigt Inhalt des Ordners an |
| cp [-r] Datei1 Datei2 Ziel | kopiert *Datei1,...* (oder mit der Option *-r* auch ganze Ordner) nach *Ziel* |
| mv Datei Ordner Ziel | verschiebt *Datei* und *Ordner* nach *Ziel* |
| cd Ordner/Unterordner | gehe zu *Ordner/Unterordner*, */* ist Wurzelverzeichnis oder Trennzeichen für Ordnernamen |
| cd .. | gehe in das übergeordnete Verzeichnis |
| rm [-r] Datei [Ordner] | *Datei* [und *Ordner*] löschen (*-r* nur bei Ordnern benutzen) |
| grep [-irn] was Dateien [Ordner] | suche *was* in *Dateien* [und *Ordner*] |
| okular Datei | zeigt Inhalt der *Datei* (.ps,.pdf,.jpg,...) an |
| gnuplot [Datei] | beginnt gnuplot oder führt den Inhalt von *Datei* aus |
| gfortran [-Wall -Wextra -pedantic -fbacktrace -fcheck=bounds -g -pg] -o test test.f90 | compiliert *test.f90* mit dem gnu compiler nach *test* mit debug Informationen |
| gfortran -c modul.f90 | compiliert *modul.f90* zu *modul.mod* |
| gcc, g++ | gnu compiler für C und C++ |
| ./Programm | *Programm* ausführen |
| <Strg> c | aktives Programm beenden |
| ps [uwx] | zeigt aktuell laufende Programme des Benutzers an |
| kill prozessid | tötet Programm *prozessid* |
| gdb Programm | lädt *Programm* mit dem gdb-Debugger, Befehle: break [Datei:]Zeile, run [Argumente für Programm], step, next, c, bt, print variable. ... |
| valgrind Programm | Debugger, gut für Speicher-Probleme |
| gprof -p Programm gmon.out > out | zeigt Performance-Daten an |
| vim Datei | öffnet *Datei* mit Konsolen-Editor *vim* |
| vimdiff Dat1, ..., Dat4 | vergleicht bis zu vier Dateien mit dem *vim* Editor |

## Vim

| | |
|---|---|
| vim [-o/-O] Datei [dat2 ...] | öffnet *Datei* [und *dat2 ...*] im *Befehls*-Modus [über / neben einander] |
| i | vom *Befehls*-Modus in den *Schreib*-Modus wechseln |
| r | ersetze einzelnes Zeichen unter dem Curser |
| R | Ersetze alle Zeichen unter dem Curser fortlaufend |
| w / b | springe zu nächstem / vorhergehenden Wort |
| <Strg>W[2]W | springe zum nächsten [zweiten] Dateifenster, sofern mehrere geöffnet sind |
| /was / ?was | sucht *irgendwas* und springt zum nächsten / vorhergehenden Treffer |
| q 1 | Aufzeichnung starten und in *1* abspeichern, Beenden der Aufzeichnung mit *q* |
| @ 1 | aufgezeichnete Abfolge von *1* ausführen |
| v / V | in den *Visuellen*-Modus wechseln und Zeichen markieren |
| I | innerhalb des *Visuellen*-Modus in den *Schreib*-Modus wechseln |
| <Esc> | in den *Befehls*-Modus zurück kehren |
| :w / :q / :wq | Speichern / Beenden / Speichern und Beenden |
| :s/alt/neu/g | ersetze alle *alt* in der Zeile durch *neu*, in der ganzen Datei mit *:%s...* |
| :split Datei / :vsplit Datei | öffne *Datei* in der unteren / rechten Hälfte des aktuellen Fensters |
| :map 1 i | eine Befehlskette in *1* abspeichern, Aufruf mit Eingabe *1* |

## Queue

| | |
|---|---|
| qsub [...] Datei | Datei mit den Optionen [...] in die Queue schicken |
| -q 32bit[@jimbo] | die Queue *32bit* oder den Computer *32bit@jimbo* auswählen |
| -N test | dem Job den Namen *test* geben |
| -cwd | gehe zu dem aktuellen Verzeichnis |
| -pe mpi 4 | benutze die parallele Umgebung *mpi* mit *4* prozessen |
| qstat [...] | zeigt alle laufenden Jobs des Benutzers an |
| -f | zeigt die gesamte Queue mit den Jobs des Benutzers an |
| -u '*' | zeigt alle laufenden Jobs aller Benutzer (*) an |
| qalter [...] jobid | verändert die Optionen des noch nicht gestarteten Jobs *jobid* |
| qdel jobid | beendet die Ausführung von *jobid* |