

- `ssh` (Secure Shell) ist ein Netzwerkprotokoll für sicheren Zugriff auf entfernte Rechner
- Der andere Rechner muss über das Netzwerk zugänglich sein und einen laufenden `ssh`-Server haben
- Der zugreifende Rechner benötigt einen `ssh`-Client
- Die Fernsteuerung erfolgt standardmäßig über Kommandozeile (Terminal)
- `scp` (Secure Copy Protocol) basiert auf `ssh` und ist für den Datenaustausch zwischen entfernten Rechnern gedacht.
- Im Rahmen dieser Vorlesung können Studierende mittels `ssh` und `scp` auf Poolrechner zugreifen: xy1234@fphctssh.physik.kit.edu

- Sehr viele Linux-Distributionen haben einen ssh- und scp-Client bereits vorinstalliert.
- Ansonsten kann man sie einfach über die Paketverwaltung nachinstallieren
 - Ubuntu/Mint/Debian: `sudo apt-get install openssh-client`
 - RedHat/CentOS/Fedora: `sudo dnf install openssh-clients`
 - Arch/Manjaro: `sudo pacman -S openssh`
- Unter macOS gehören ssh und scp zu den Bordmitteln. Sie können über das Terminal gestartet werden
- Unter Windows 10 sind ssh und scp ab der Ausgabe 1809 direkt verfügbar. Man kann auf sie bequem über die Power Shell zugreifen
- Frühere Windows-Versionen benötigen Hilfsprogramme wie [Putty](#) oder [WinSCP](#)

- In C++ liegt der Quellcode in Form von menschenlesbaren Textdateien vor. Meist haben diese die Endungen `.cpp` und `.h`
- Diese Dateien sind aber nicht direkt ausführbar und müssen zuerst in die Maschinsprache übersetzt (compiliert) werden.
- Der Übersetzer (Compiler) erzeugt eine ausführbare Datei (Binary), mit der das Programm gestartet werden kann
- Die Binaries müssen in der Regel zum Betriebssystem (z.B. Windows) und Rechnerarchitektur (z.B. x86 64-bit) passen
- Verbreitete quelloffene Compiler sind `gcc` (GNU Compiler Collection) und `clang`, welches als Frontend zu LLVM (Low Level Virtual Machine) fungiert
- Es gibt aber auch proprietäre Compiler, wie z.B. Visual C++ von Microsoft oder Intel C++ Compiler

- Ein Compiler übersetzt den vorliegenden Quellcode und gibt Fehlermeldungen bei Syntaxfehlern aus
- Zum Schreiben von Quellcode braucht man aber mindestens einen Editor
- Viele Editors beherrschen (direkt oder über Erweiterungen) Syntaxhervorhebung und Codevervollständigung, wodurch das Programmieren schneller und angenehmer wird
- Wer noch mehr Komfortfunktionen haben möchte, benutzt eine integrierte Entwicklungsumgebung oder IDE (Integrated Development Environment)

- Unter Linux setzt man vorwiegend auf `gcc`, der Compiler lässt sich bequem über die Paketverwaltung nachinstallieren
 - Ubuntu/Mint/Debian: `sudo apt-get install g++`
 - RedHat/CentOS/Fedora: `sudo dnf install gcc-c++`
 - Arch/Manjaro: `pacman -Sy gcc`
- Es gibt zahlreiche Editors wie `vim`, `emacs`, `gedit` oder `kate` mit denen man effizient in C++ entwickeln kann
- Auch quelloffene IDEs für C++ sind reichlich vorhanden: `Geany`, `Gnome Builder`, `KDevelop`, `Eclipse CDT`, `QT Creator`, `Code::Blocks`, `Visual Studio Code`

- macOS Nutzer können ihre C++ Quelldateien mit `clang` übersetzen
- Beim ersten Aufruf von `clang++` im Terminal wird der Compiler automatisch nachinstalliert.
- Apple bietet auch eine kostenlose IDE an: `Apple Xcode`
- Unter den neuesten macOS Versionen kann man `Xcode` bequem über den `App Store` installieren
- Benutzer von älteren Versionen müssen sich eine passende `Xcode`-Ausgabe (z.B. `Xcode 10.1` für High Sierra) bei `Apple Developer` heraussuchen.
- Alternativ besteht immer die Möglichkeit, eine andere IDE wie z.B. `Eclipse CDT`, `QT Creator` oder `Visual Studio Code` zu installieren

- Unter Windows muss ein C++ Compiler auf jeden Fall per Hand nachinstalliert werden.
- Wer bei Microsoft bleiben möchte, kann sich das kostenlose **Visual Studio Community** (IDE+Compiler) herunterladen.
- Auch die quelloffene IDE **Orwell Dev-C++** enthält einen Compiler (gcc)
- **Code::Blocks** ist ebenfalls OpenSource und leichtgewichtig. Die Installationsdatei `codeblocks-20.03mingw-setup.exe` enthält bereits einen C++ Compiler (MinGW-Variante von gcc).

- Der Kernsprache C++ fehlen viele Funktionen, die man im Alltag häufig braucht.
- Diese können aber über Bibliotheken (Libraries) nachgeladen werden.
- Abgesehen von zahlreichen externen Bibliotheken gibt es auch die sog. C++ Standardbibliothek
- Die Standardbibliothek ist standardisiert und gehört zum offiziellen Sprachumfang von C++
- Die Unterbibliothek `cmath` enthält dabei wichtige mathematische Funktionen, die im nackten C++ nicht vorhanden sind.
- Dazu gehört die Exponentialfunktion (`exp`) aber auch Logarithmen (`log`, `log10`) oder trigonometrische Funktionen (`sin`, `cos`, `tan` usw.)

- Felder (Arrays) sind ein wichtiges Instrument, um verschiedene Arten von Daten im Programm effizient zu organisieren und zu verarbeiten.
- Es ist stets zu beachten, dass ein frisch definiertes Feld in der Regel nicht leer ist
- Zugriff auf undefinierte Felder wird an sich nicht unterbunden
- Man kann aber neue Felder gleich mit Nullen füllen, in etwa so

```
#include <iostream>

int main () {

int vec[3] = {0};

std::cout << vec[0] << std::endl;
std::cout << vec[1] << std::endl;
std::cout << vec[2] << std::endl;

return 0;
}
```

- Der Zugriff auf nichtexistierende Elemente eines Felds ist ebenfalls möglich und führt zum sog. undefinierten Verhalten

```
#include <iostream>

int main () {

int vec[3] = {0};

std::cout << vec[3] << std::endl; // illegal!

return 0;
}
```

- Wird ein Feld bereits bei der Definition vollständig initialisiert, so kann man die Anzahl der Elemente auch auslassen

```
#include <iostream>

int main () {

int vec[] = {10, 20, 30};

std::cout << vec[0] << std::endl;
std::cout << vec[1] << std::endl;
std::cout << vec[2] << std::endl;

return 0;
}
```