Paul Nykiel

С

Ein erstes ( Programm

Buildproze

Speicher

1 011110

\_\_\_\_

Ein erstes

Programm

Programm

Zero-Copy

# Eine Einführung in modernes C++ Teil 1 – Basics

Paul Nykiel

28. April 2020

Speicher

Pointer

Ein erstes C Programm

**Buildprozess** 

Speicher

Pointer

6 C++

Ein erstes C++ Programm

Paul Nykiel

С

Ein erstes C Programm Buildprozess

Speicher

Pointer

. .....

Ein erstes

Programm

Zero-Copy

C

C

Ein erstes ( Programm

Buildprozess

Speicher

Point

Ein erstes

C++ Programm

Zero-Copy

• ANSI/ISO Standardisierte Programmiersprache

Ein erstes

Programm
Buildprozess

Speicher

Pointe

C++

Fin erste

C++ Programm

- ANSI/ISO Standardisierte Programmiersprache
- Entwickelt ab 1969 für Unix

C

Programm

Buildprozess

Speicher

Point

Ein erstes

C++ Programm

- ANSI/ISO Standardisierte Programmiersprache
- Entwickelt ab 1969 für Unix
- "Portabler Assembler"

.

Ein erstes C Programm

Buildproz

Speiche

Point

C++

Ein erstes

C++ Programm

- ANSI/ISO Standardisierte Programmiersprache
- Entwickelt ab 1969 für Unix
- "Portabler Assembler"
- Einfach zu verstehen, schwierig zu verwenden

Buildprozess

Speicher

Pointer

C++

Ein erstes

C++ Programm

Zero-Copy

## C im Vergleich zu Java

• Undefiniertes Verhalten

Buildprozess

Speicher

.

Pointer

- ' '

Ein erstes C++

Zero-Copy

## C im Vergleich zu Java

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung

Buildproz

Speiche

.

Point

C+-

Ein erste

C++ Programm

Zero-Copy

## C im Vergleich zu Java

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung
- Kleiner Sprachkern und kleine Standardlibrary

Paul Nykiel

Ein erstes C Programm

Buildprozess

Speicher

Pointer

- ' '

Ein erstes

Programm

Zero-Copy

## Ein erstes C Programm

Paul Nykiel

C

Ein erstes C Programm

Buildprozess

Speicher

----

Pointer

C++

Ein erstes

C++ Programm

Zero-Copy

# Beispiel: Hello World

Paul Nykiel

C.

Ein erstes ( Programm

Buildprozess

Speicher

Pointer

0 1 1

Ein erstes

Programm

Zero-Copy

## Build prozess

Buildprozess

Speicher

Point

C+-

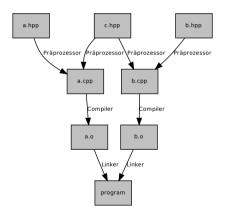
Ein erstes

Programn

Zero-Copy

#### Vom Sourcecode zur ausführbaren Datei

Präprozessor



Buildprozess

Speicher

Dalak

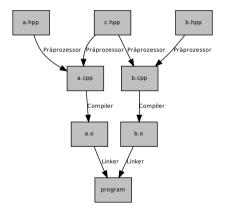
Ein aust

C++ Programn

Zero-Copy

### Vom Sourcecode zur ausführbaren Datei

- Präprozessor
- Compiler



Buildprozess

Speicher

. 01111

C . .

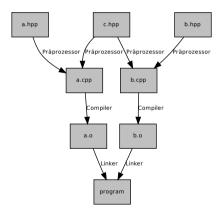
Ein erstes

C++ Programm

Zero-Copy

#### Vom Sourcecode zur ausführbaren Datei

- Präprozessor
- Compiler
- Linker



Buildprozess

Speiche

Dalah

. .....

C++

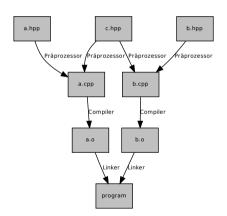
Ein erste

C++ Programn

Zero-Copy

#### Vom Sourcecode zur ausführbaren Datei

- Präprozessor
- Compiler
- Linker
- #includes sichern Typkonsistenz



Paul Nykiel

C

Ein erstes C Programm

Buildprozess

Speichei

Dointor

. .....

Ein orstos

C++

Zero-Copy

# Beispiel: Eine zweite Übersetzungseinheit

#### Buildprozess

Speicher

Pointer

Ein erstes

C++ Programm

Zero-Copy

## Warum ein Buildsystem

• Nur geänderte Dateien neu kompilieren

#### Build prozess

Speiche

Point

Fin erste

C++ Programm

Zero-Copy

## Warum ein Buildsystem

- Nur geänderte Dateien neu kompilieren
- Einzelner Befehl an Compiler wird zu kompliziert

#### Buildprozess

Speiche

ъ.

. ....

C++

Ein erstes C++

Zero-Copy

## Warum ein Buildsystem

- Nur geänderte Dateien neu kompilieren
- Einzelner Befehl an Compiler wird zu kompliziert
- Portabilität

```
CMake
 modernes
  C++
          Wird durch Datei CMakeLists.txt konfiguriert:
Paul Nykiel
          cmake_minimum_required(VERSION 3.10)
          set(STANDARD 17)
Buildprozess
          set(COMMON_FLAGS "-Wall")
          if (CMAKE_BUILD_TYPE STREQUAL "Release")
               set(COMMON_FLAGS "${COMMON_FLAGS} -03")
          endif ()
          set(CMAKE C STANDARD ${STANDARD})
          set(CMAKE CXX STANDARD ${STANDARD})
          set(CMAKE C FLAGS ${COMMON FLAGS})
          set(CMAKE_CXX_FLAGS ${COMMON_FLAGS})
          project(MyProject)
                                                              4 D > 4 D > 4 E > 4 E > E 990
```

Eine Einführung in

C

Ein erstes C Programm

Buildprozess

Speicher

Pointer

C++

Programm

Zero-Copy

# Beispiel: CMake

Paul Nykiel

C.

Ein erstes ( Programm

Buildprozess

Speicher

Pointer

C++

Ein erstes

Programm

Zero-Copy

Speicher

Ein erstes C

Buildprozess

#### Speicher

\_\_\_\_

FOIIL

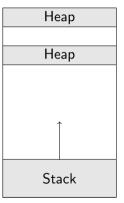
\_\_\_\_\_

C++

Programm

### Stack und Heap

 Hauptspeicher (RAM) wird aus zwei Richtungen vergeben



Buildprozess

Speicher

0 00.0..

Pointe

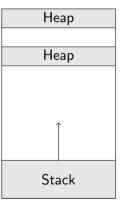
\_.

C++

Programm

Zero-Copy

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut



Buildprozess

#### Speicher

Pointe

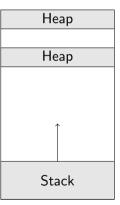
. .....

Ein orete

C++

Zero-Copy

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut
- Heap für dynamischen Speicher



Ein erstes C

Buildprozes

#### Speicher

Pointe

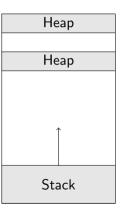
- -----

- ' '

Ein erstes

Zero-Cop

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut
- Heap für dynamischen Speicher
- Speicher auf dem Heap muss händisch reserviert und freigegeben werden



Buildprozes

#### Speicher

Pointe

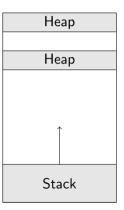
. 01116

C++

Ein erstes C++

Zero-Con

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut
- Heap für dynamischen Speicher
- Speicher auf dem Heap muss händisch reserviert und freigegeben werden
- Bei mehr als einem Owner Verwaltung kompliziert



Ein erstes ( Programm

Buildprozess

Speicher

Pointe

Ein erstes C++

Programm
Zero-Conv

```
int sqr(int arg) {
    int res = arg * arg;
    return res;
int main(void) {
    int a = 17;
    int b = sqr(a);
    return 0;
```

```
Paul Nykiel
```

Buildprozess

#### Speicher

```
int main(void) {
    int a = 17;
    int b = sqr(a);
    return 0;
}
```

int sqr(int arg) {

return res;

int res = arg \* arg;

```
int b
int a
```

Ein erstes

Buildprozess

Speicher

Dainta

romter

Ein erstes

C++ Programm

```
int sqr(int arg) {
    int res = arg * arg;
    return res;
int main(void) {
    int a = 17;
    int b = sqr(a);
    return 0;
```

```
Verwaltungsinfo
    int b
    int a
```

C

Ein erstes ( Programm

Buildprozess

Speicher

Pointe

C . .

Ein erstes

C++ Programm

```
int sqr(int arg) {
    int res = arg * arg;
    return res;
int main(void) {
    int a = 17;
    int b = sqr(a);
    return 0:
```

```
int arg
Verwaltungsinfo
    int b
    int a
```

```
Paul Nykiel
```

Programm

Buildprozess

#### Speicher

Pointe

Officer

Ein erstes

C++ Programm

```
int sqr(int arg) {
    int res = arg * arg;
    return res;
int main(void) {
    int a = 17;
    int b = sqr(a);
    return 0:
```

```
int res
   int arg
Verwaltungsinfo
    int b
    int a
```

```
Paul Nykiel
```

Buildprozess

Speicher

```
int sqr(int arg) {
    int res = arg * arg;
    return res;
int main(void) {
    int a = 17;
    int b = sqr(a);
    return 0;
}
```

int b int a

C

Ein erstes ( Programm

Buildprozess

#### Speicher

Point

Pointe

Fin oretos

C++

Zero-Copy

# Reicht das nicht?

Buildprozess

Speicher

Pointe

C++

Ein erstes

Programm

ero-Copy

```
int main(void) {
    int *a = new int[2];
    int *b = new int[2];
    delete[] a;
    return 0;
}
```

In Verwendung

C

Ein erstes C Programm

Buildprozess

Speicher

Pointe

\_

Ein erstes

C++ Programm

ero-Copy

```
int main(void) {
    int *a = new int[2];
    int *b = new int[2];
    delete[] a;
    return 0;
}
```

In Verwendung

а

С

Ein erstes ( Programm

Buildprozess

Speicher

Pointe

C++

Ein erstes

C++ Programm

ro-Copy

```
int main(void) {
    int *a = new int[2];
    int *b = new int[2];
    delete[] a;
    return 0;
}
```

In Verwendung

a

b

Buildprozess Speicher

\_\_\_\_\_\_

Pointe

Ein orstos

C++ Programm

Zero-Copy

```
int main(void) {
    int *a = new int[2];
    int *b = new int[2];
    delete[] a;
    return 0;
}
```

In Verwendung

....verwendung

b

Paul Nykiel

(

Ein erstes C Programm

Buildprozess

Speicher

#### Pointer

C++

Ein erstes

Programm

Zero-Copy

Pointer

• Pointer

Ein erstes C Programm

Buildprozess

Speicher

Speichei

Pointer

C++

Ein erstes

Programm

Zero-Copy

Ein erstes (

Programm Buildprozess

Speicher

operence

Pointer

Ein erstes

C++ Programm

- Pointer
- Angst!

Ein erstes

Programm Buildprozess

Speicher

operene

Pointer

Ein exetee

C++

Programm
Zero-Copy

- Pointer
- Angst!
- Gefährlich!

Programm
Buildprozess

Speicher

Operene

Pointer

C++

Ein erstes C++ Programm

- Pointer
- Angst!
- Gefährlich!
- Böse!

Programm Buildprozess

Speicher

Pointer

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential

C

Ein erstes C Programm

Buildprozess

Speiche

Pointer

CTT

Ein erstes C++

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- Was ist ein Pointer: eigentlich nur ein int den der Compiler separat behandelt.

Buildprozes

Speiche

#### Pointer

Ein erstes

Programm

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- Was ist ein Pointer: eigentlich nur ein int den der Compiler separat behandelt.
- int b = 17; int \*a = &b;

Speicher

Pointer

#### Oiiic

Ein erstes

C++ Programn

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- Was ist ein Pointer: eigentlich nur ein int den der Compiler separat behandelt.
- int b = 17; int \*a = &b;
- int \*c = new int();

Programm

Speiche

Pointer

#### OIIILE

Ein erstes

C++ Programn

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- Was ist ein Pointer: eigentlich nur ein int den der Compiler separat behandelt.
- int b = 17; int \*a = &b;
- int \*c = new int();
- delete c;

Programm

Speiche

#### Pointer

Ein erste

C++ Programn

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- Was ist ein Pointer: eigentlich nur ein int den der Compiler separat behandelt.
- int b = 17; int \*a = &b;
- int \*c = new int();
- delete c;
- Speicherverwaltung gehört nicht in die Anwendungslogik

```
Paul Nykiel
```

Ein erstes (

Programm
Buildprozess

Speicher

-----

#### Pointer

C++

Ein erstes C++ Programm

```
int main() {
   int a = 0; // Liegt auf dem Stack
   int *aPtr = &a; // Zeigt auf den Stack

   int *bPtr = new int(); // Liegt auf dem Heap
   delete bPtr; // Speicher muss freigegeben werden
}
```

Paul Nykiel

Ein erstes ( Programm

Buildprozess

Speicher

Pointer

C++

Ein erstes

Programm

Zero-Copy

C++

Programm

Buildprozess

Speicher

C++

Ein erstes

Programm

Zero-Copy

• C with classes

Fin erstes

Programm

Buildprozess

Speicher

Pointer C++

Ein erstes

C++

- C with classes
- Wird in quasi jeder Domäne genutzt

\_

Ein erstes ( Programm

Buildprozess

Speicher

Pointer

C++

Ein erste

C++ Programm

- C with classes
- Wird in quasi jeder Domäne genutzt
- Ziele: Performanter und sicherer Code

Speicher

Opererie

Pointer

C++

C | |

Ein erstes C++

Zero-Copy

C++ im Vergleich zu Java

Templates

Fin orstos

Programm
Buildprozess

Speicher

-----

Point

C++

Ein erstes

C++ Programm

- Templates
- Operatorenüberladung

Buildproze

Speicher

Pointe

C++

Ein erste

C++ Programm

Zero-Copy

# C++ im Vergleich zu Java

- Templates
- Operatorenüberladung
- Tendentiell weniger tiefe Vererbung

Buildprozes

Speiche

Pointe

C++

Ein erste

C++ Programn

Zero-Copy

### C++ im Vergleich zu Java

- Templates
- Operatorenüberladung
- Tendentiell weniger tiefe Vererbung
- Mehrfachvererbung

Buildproze

Speiche

Pointe

C++

Fin erste

C++
Programs

- -

Zero-Copy

### C++ im Vergleich zu Java

- Templates
- Operatorenüberladung
- Tendentiell weniger tiefe Vererbung
- Mehrfachvererbung
- Definierte Objektlebenszeit

Paul Nykiel

C

Ein erstes ( Programm

Buildprozess

Speicher

Pointer

Ein erstes C++

Programm

Zero-Copy

Ein erstes C++ Programm

Paul Nykiel

C

Ein erstes C Programm

Buildprozess

Speicher

o porono.

Pointer

C++

Ein erstes

C++ Programm

Zero-Copy

# Beispiel: Hello World

Paul Nykiel

Programm

Buildprozess

Speicher

Pointer

Programm

Zero-Copy

C

Ein erstes C Programm

Buildprozess

Speicher

Point

\_\_\_

C++ Programm

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

Buildproze

Speiche

Dointo

- -----

Ein erstes

C++ Programm

Zero-Copy

# Speicherverwaltung

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

• Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente

Buildproze

Speiche

Pointe

Ein erstes

C++ Programm

Zero-Copy

# Speicherverwaltung

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

- Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente
- Einfach verständlich

Buildproz

Speicne

Point

Ein erstes C++

C++ Programm

Zero-Copy

# Speicherverwaltung

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

- Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente
- Einfach verständlich
- Für große Objekte unnötige Performanceeinbuße

Programm Buildprozess

Speicher

Pointer

Zero-Copy

• Standardlibrary kann Verwaltung übernehmen

Buildprozess

Speicher

1 OIIIC

- ' '

C++
Programm

Zero-Copy

### Smart-Pointer

- Standardlibrary kann Verwaltung übernehmen
- unique\_ptr

Ein au

Programm

Speicher

Opererie

Pointe

C++

Ein erste

C++ Programm

- Standardlibrary kann Verwaltung übernehmen
- unique\_ptr
- Genau ein Owner

Speiche

Pointe

C + +

Ein erstes C++

Zero-Copy

#### Smart-Pointer

- Standardlibrary kann Verwaltung übernehmen
- unique\_ptr
- Genau ein Owner
- std::unique\_ptr<int> a = std::make\_unique<int>(17);

Programm

Puildarazasa

Speiche

Fointe

Ein erstes

C++ Programm

- Standardlibrary kann Verwaltung übernehmen
- unique\_ptr
- Genau ein Owner
- std::unique\_ptr<int> a = std::make\_unique<int>(17);
- $\bullet$  shared\_ptr

\_ . .

Speiche

Point

E-------

Ein erstes C++ Programm

Zero-Copy

#### Smart-Pointer

- Standardlibrary kann Verwaltung übernehmen
- unique\_ptr
- Genau ein Owner
- std::unique\_ptr<int> a = std::make\_unique<int>(17);
- shared\_ptr
- Quasi immer nutzbar
- std::shared\_ptr<int> a = std::make\_shared<int>(17);

Buildprozess

Speicher

Pointer

C++

Ein erstes C++

Programm

Zero-Copy

### Referenzen

• Sprachfeature kein Library-Feature

Ein erstes

Programm

Buildprozess

Speicher

Pointer

1 011110

Ein erste

C++ Programm

- Sprachfeature kein Library-Feature
- Können nicht null sein

Fin eretes

Programm

Buildprozess

Speicher

Point

Ein erste

C++ Programm

- Sprachfeature kein Library-Feature
- Können nicht null sein
- Können aber ungültig werden

Buildproze

Speicher

Dalah

Ein erste

C++ Programm

- Sprachfeature kein Library-Feature
- Können nicht null sein
- Können aber ungültig werden
- int b = 17; int &a = b;

Buildprozess

Speicher

Point

C . .

Ein erstes

 $\begin{array}{c} C++ \\ Programm \end{array}$ 

Zero-Copy

# Zusammenfassung Pointer

• Raw-Pointer: Sollten quasi nie verwendet werden

Buildproze

Speicher

Dainta

. .....

Ein erste

C++ Programm

Zero-Copy

# Zusammenfassung Pointer

• Raw-Pointer: Sollten quasi nie verwendet werden

• Unique-Pointer: Oftmals ersatz für Raw-Pointer

Buildproze

Speiche

Point

\_

Ein erstes C++

C++ Programm

Zero-Copy

# Zusammenfassung Pointer

• Raw-Pointer: Sollten quasi nie verwendet werden

• Unique-Pointer: Oftmals ersatz für Raw-Pointer

Shared-Pointer: Sichere Pointer f
ür beliebig viele Owner

Builaproz

Speiche

Point

Ein erstes

C++ Programm

Zero-Copy

### Zusammenfassung Pointer

- Raw-Pointer: Sollten quasi nie verwendet werden
- Unique-Pointer: Oftmals ersatz f
  ür Raw-Pointer
- Shared-Pointer: Sichere Pointer für beliebig viele Owner
- Referenzen: Oftmals um Kopien zu vermeiden

C

Ein erstes C Programm

Speicher

0 10 10 110 1

Pointe

C++

Ein erstes

Programm

Zero-Copy

# Beispiel: Pointer & Referenzen