

Systemnahe Programmierung mit C++

Stephan Neuhaus

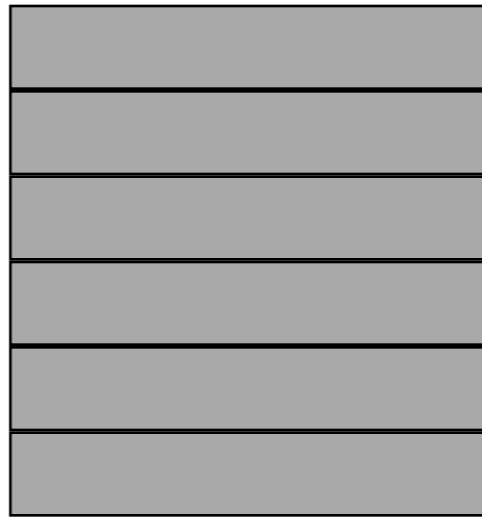
HISPOS-Anmeldung

“bitte denken Sie daran, die Teilnehmer Ihrer Veranstaltungen darauf hinzuweisen, dass sie sich bis zum 22. Juni bei HISPOS für die Veranstaltung anmelden müssen, um Leitungspunkte für die Veranstaltung bekommen zu können.”

Gert Smolka
Studienbeauftragter

Speicher

1 byte = 8 bits



0x00000000
0x00000001
0x00000002
0x00000003
0x00000004
0x00000005

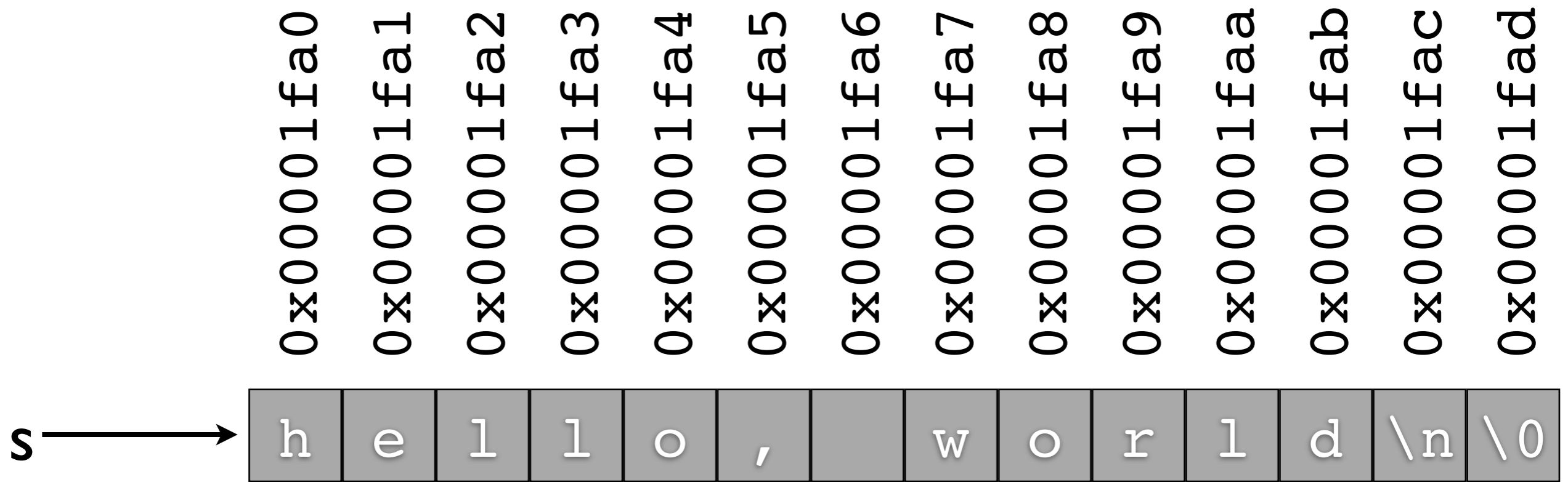
...



0xfffffffffa
0xfffffffffb
0xfffffffffc
0xffffffffffd
0xffffffffffe
0xfffffffffff

Speicher

```
char* s = "hello, world\n";
```



Demo: Speicher

Konzepte

- Speicher-Layout
- Adresse (= Pointer)

Structs

```
class TreeNode {  
public:  
    char* info;  
    TreeNode* left;  
    TreeNode* right;  
};
```

Structs

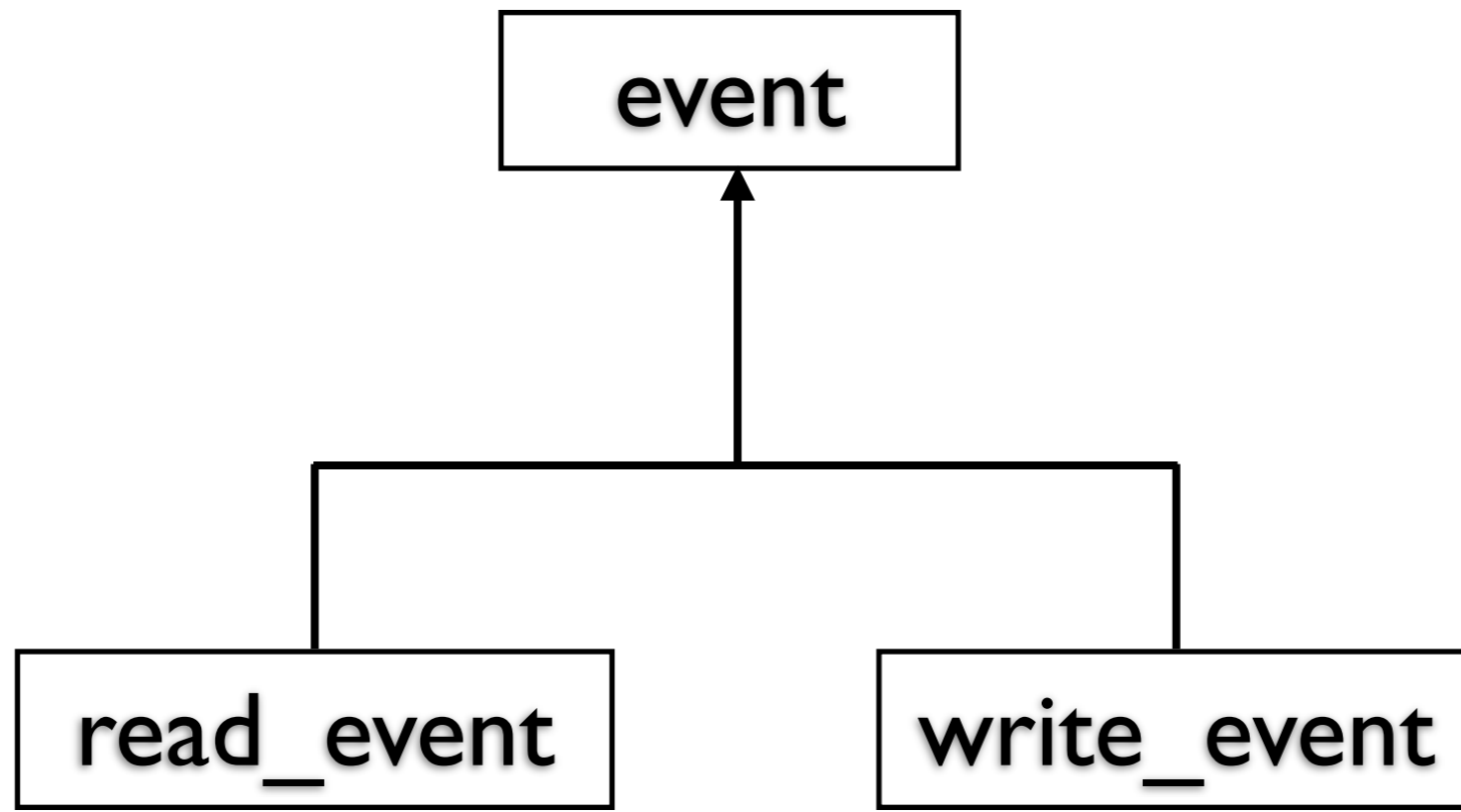
```
struct treenode_t {  
  
    char* info;  
    treenode_t* left;  
    treenode_t* right;  
};
```


Demo: Structs

Unions

- Vererbung für Arme
- Teilen sich denselben Speicherplatz

Demo: Unions



Konzepte

- Member-Zugriff
- Struct-Deklaration
- Forward-Declaration
- Structs nur als einfache Datencontainer
- Unions: Vererbung für Arme

Arrays und Pointer

```
struct X {  
    int count;  
    double x;  
};  
  
X array[4];
```

0	1	2	3
count	count	count	count
x	x	x	x
0x00002020	0x0000202c	0x00002038	0x00002044

Demo: Arrays

Konzepte

- Arrays
- Sizeof
- Pointerarithmetik
- Äquivalenz von Arrays und Pointern

Makros

- “const” für Arme
- Inline-Funktionen für Arme

Demo: Makros

String-Copy

```
char* strcpy(char* dst, const char* src) {  
    while (*dst != '\0') {  
        *dst = *src;  
        src++;  
        dst++;  
    }  
  
    *src = *dst;  
  
    return dst;  
}
```

String-Copy

```
char* strcpy(char* dst, const char* src) {  
    while (*dst != '\0') {  
        *dst++ = *src++;  
    }  
  
    *dst = *src;  
  
    return dst;  
}
```

String-Copy

```
char* strcpy(char* dst, const char* src) {  
    while ((*dst++ = *src++) != '\0')  
        ; // empty  
  
    return dst;  
}
```

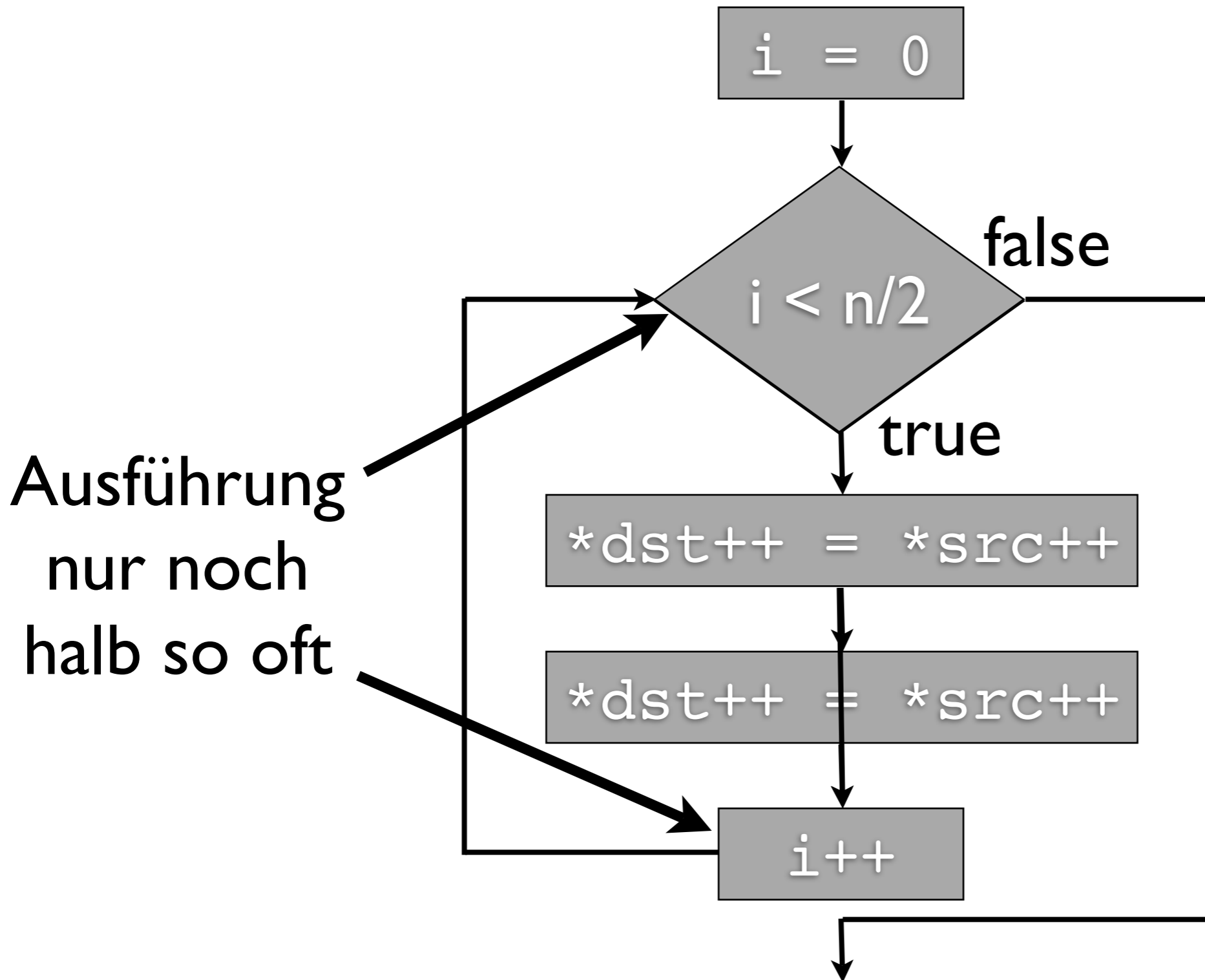
String-Copy

```
char* strcpy(char* dst, const char* src) {  
    while (*dst++ = *src++)  
        ; // empty  
  
    return dst;  
}
```

Duff's Device

- Beschleunigung durch Loop Unrolling
- In der Vorlesung nicht behandelt!

Unrolling



Demo: Unrolling

Konzepte

- Unrolling
- Standardbibliothek nutzen!

Konzepte

- Speicher: Abfolge von Bytes mit Adresse
- Arrays und Pointer: Äquivalent durch Arithmetik
- Unions: Arme-Leute-Vererbung
- Makros: Arme-Leute-const und -inline