

# KV Betriebssysteme

## Einführung in C

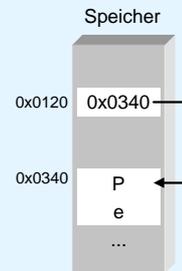
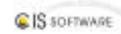
### KV Betriebssysteme

### Zeiger

DI. Dr. Peter René Dietmüller

CIS-Software - Peter René Dietmüller  
Lorcherstraße 2a, 4470 Enns

### Zeiger - Modell



### Zeiger (I)



#### ↳ Deklaration eines Zeigers:

- BaseType "\*" Identifier

#### ↳ Beispiele:

- `int *zeiger1;`
- `int *zeiger2, wert1; /* 'wert 1' ist kein Zeiger! */`

#### ↳ Adresse einer Variablen:

- &-Operator

#### ↳ Beispiele:

- `zeiger1 = &wert1;`

### Zeiger (II)



#### ↳ Dereferenzierung eines Zeigers

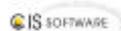
- \*-Operator

#### ↳ Beispiel

- `int iValue;`
- `int *piValue;`

```
iValue = 100;
*piValue = 200;
printf("%d\n", Value); /* ?? */
printf("%X\n", piValue); /* ?? */
printf("%d\n", *piValue); /* ?? */
```

### Zeigerarithmetik



#### ↳ Addition zu Zeigern entspricht Adress-Addition von `n*sizeof(element)`

- `int *zeiger, wert; zeiger = &wert; zeiger++;`

#### ↳ Subtraktion zu Zeigern entspricht Adress-Subtraktion von `n*sizeof(element)`

- `int *zeiger, wert; zeiger = &wert; zeiger--;`

#### ↳ Differenzbildung von zwei Zeigern auf Arrays liefert die Anzahl der dazwischenliegenden Elemente

- `int *zeiger1, *zeiger2; ...;`
- `printf(„,%d“, zeiger2 - zeiger1);`

### Zeiger und Arrays

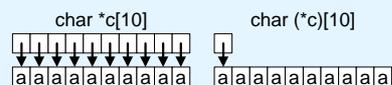


#### ↳ Verwendung von `arrayName` entspricht der Adresse des ersten Arrayelementes.

- `int werte[10]; /* werte == &werte[0] */`

#### ↳ Unterschied:

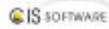
- `char *c[10];` ... array of pointers
- `char (*c)[10];` ... pointer to array



# KV Betriebssysteme

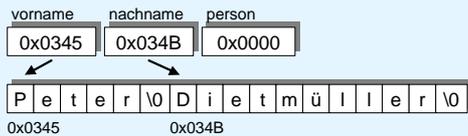
## Einführung in C

### Zeiger und Zeichenketten

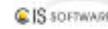


↳ In C sind Zeichenketten Character-Arrays, die mit einem Nullbyte abgeschlossen werden.

- char vorname[] = „Peter“;
- char \*nachname = „Dietmüller“;
- char \*person;

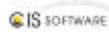


### Zeiger - Beispiele (I)



```
/* -- Unterschiedliche Deklarationen -- */
#include <stdio.h>
int main (void) {
    char *a[10],*(b[10]),(*c)[10];
    printf("%d %d %d\n",
           sizeof(a),sizeof(b),sizeof(c));
    /* Ergebnis: 40, 40, 4 */
    return 0;
}
```

### Zeiger - Beispiele (II)



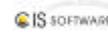
↳ Anhängen eines Zeichens an eine Zeichenkette:

```
while (*c) c++;
*c++='!';
*c=0;
```

↳ Dasselbe mit einer Funktion aus <string.h>

```
strcat(c, "!");
```

### Zeiger - Beispiele (III)



↳ Zeigerarithmetik:

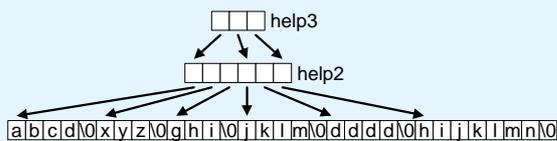
```
int i, *p, fib[] = {1,1,2,3,5,8};
```

```
/* -- Ausgabe des Arrays in drei Varianten -- */
for (i = 0; i <= 5; i++) printf("%d\n", fib[i]);
for (p = &fib[0]; p <= &fib[5]; p++)
    printf("%d\n", *p);
for (p = fib; p <= fib+5; p++) printf("%d\n", *p);
```

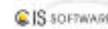
### Zeigerarithmetik - Beispiel (I)



```
char *string = "abcdefghijklmnopqrstuvwxy";
char *help2[] = { "abcd", "xyz", "ghi",
                 "jklm", "ddd", "hijklmn" };
char **help3[] = { help2, help2+3, help2+5 };
```



### Zeigerarithmetik - Beispiel (II)



```
int main (void) {
```

```
    char *help,**help4;
```

```
    help = string+10;
```

```
    printf("%s\n", help); /* klmnopqrstuvwxyz */
```

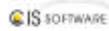
```
    printf("%s\n", help-3); /* hijklmnopqrstuvwxyz */
```

```
    printf("%s\n\n", help+5); /* pqrstuvwxyz */
```

# KV Betriebssysteme

## Einführung in C

### Zeigerarithmetik - Beispiel (III)



```
printf("%s\n", *(help2+3)); /* jklm */
printf("%s\n", help2[5]); /* hijklmn */
printf("%s\n", help2[1]+2); /* z */

printf("%s\n", **help3); /* abcd */
printf("%s\n", *help3[1]); /* jklm */
printf("%s\n", ***(help3+2)); /* hijklmn */
```

### Zeigerarithmetik - Beispiel (4)



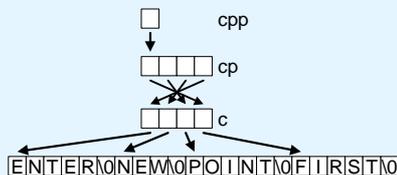
```
help4 = help3+1;
printf("%s\n", *++*help4); /* dddd */
printf("%s\n", *++(*help4++)); /* hijklmn */
printf("%s\n", *--*help4+2); /* dd */

return 0;
}
```

### Zeigersalat (I)



```
char *c[]={ "ENTER", "NEW", "POINT", "FIRST" };
char **cp[]={ c+3, c+2, c+1, c };
char ***cpp= cp;
```



### Zeigersalat (II)

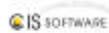


```
int main (void) {

printf("%s", **++cpp); /* POINT */
printf("%s", *--*++cpp+3); /* ER */
printf("%s", *cpp[-2]+3); /* ST */
printf("%s\n", cpp[-1][-1]+2); /* EW */

return 0;
}
```

### Call by Reference (I)



↳ C unterstützt nur die Call-by-Value Übergabe.  
Durch Zeiger kann aber Call-by-Reference simuliert werden:

```
void CallByValue (int i) {
    i=1;
}

void CallByReference (int *i) {
    *i=2;
}
```

### Call by Reference (II)



```
int main (void) {

int n = 3;
CallByValue(n);
printf("%d\n", n); /* 3 (nicht 1!!!) */
CallByReference(&n);
printf("%d\n", n); /* 2 */

return 0;
}
```