

# Betriebssysteme 1

Ebene: Nutzer – Anwendungen – Betriebssystem – Hardware

## 1. definition

- eine Sammlung von Software
- die die *Hardwareressourcen* eines Rechensystems verwaltet
- machen die Hardwareressourcen *auf geeignete Weise* verfügbar für die Anwendungen

## 2. Abstraktion

- CPU : Prozesse/Thread
- Speicher : Adressraum
- Festplatte : Dateien

## 3. die Geschichte von Betriebssysteme

- Mainframe-Ära -> Unix(Betriebssysteme im engeren Sinne) -> Personal Computer (Windows/ MacOS/Linux) -> Moderne Betriebssysteme (Linux, MacS X, iOS...)

## 4. Ziel der Vorlesung

- Betriebssysteme entwickeln, verändern oder verwalten
- Performance eines Systemes verstehen

## 5. Struktur der Vorlesung:

- Virtualisierung
- Nebenläufigkeit
- Persistenz
- Sicherheit

## 6. C crash-course

- `int x;` in Java, x ist garantiert 0, while in C, x ist nicht definiert
- *forward declaration* tells the compiler the existence of a function before its definition

```
void otherFun(void); /* forward declaration to tell compiler about otherFun */
int main() {
    int myVar = 0;
    short myOtherVar = 5;
    /* declared below */
    otherFun();
    return 12; }
void otherFun(void) {
    /* do something */
}
```

- *Pointer*: eine Variable. Enthält eine Referenz(zu einer anderen Variablen oder Datenstruktur). In C sind Pointer explizit und entsprechen der Adresse im Speicher

```
/* declaration*/
int *pointer1, *pointer2;

/* definition */
int var;
pointer1 = &var; // Pointer auf var

/* dereference*/
*pointer1 = 5; // sign 5 to the variable var, the value of var now becomes 5
```

Fix bug:

```
printf("Value of var: %d", var); // output 5%
printf("Value of var: %d\n", var); // correct output 5
```

默认使用缓冲区，若缺少换行符，可能使输出无法立即显示导致输出混乱

- Arraylänge ist in C zur Laufzeit unbekannt. C 中静态数组长度已知，对于动态数组，必须手动管理内存大小。
- String in C: null-terminierte Arrays von Zeichen des Typs char

```
char name[] = "LMU"  
// erzeuge ein passendes lokales Array und kopiere die angegebene String-Konstante  
hinein"
```

C 语言中所谓的 **String** 是 Char 型数组. End with 0

- Some basic functions:

```
puts("Hello");  
printf("The number of %s is %d", name, s);  
scanf("%d", &num);
```

print formatted: %d, %f, %s scan formatted is similar to print formatted, but read from the input and save the value in the result.

```
int strcmp(s1,s2); // compare two string, return -1, 0, 1 back to its lexico_order  
size_t strlen(s);  
char *strcpy(s1,s2); // copy from s2 to s1 with "\0", return s1  
char *strcat(s1,s2); // konkateniert s2 an s1, return s1
```

String wachsen nicht automatisch, **Puffer müssen groß genug** gewählt werden.