

Betriebssysteme - Übungen

const versus #define in C

Franz Wiesinger

Hardware-Software-Design
FH Hagenberg

Vorteile von const

- ⇒ Mit **const** wird **fester Speicherplatz** zugewiesen, auf den auch ein Pointer zugreifen kann.
- ⇒ Die Konstante wird **nur einmal im Speicher gehalten**, wogegen Konstanten per **#define** bei jedem Aufruf wieder als Wert im Opcode auftauchen.
- ⇒ Wird die Konstante mit **const im RAM-Bereich** ablegt, kann man sie **beim Debuggen verändern**, wenn man möchte.

Ein **#define** ist ein Makro, das vor dem eigentlichen Compilerlauf (vom Präprozessor) im Source-Code mit dem Wert ersetzt wird, der im **#define** angegeben ist.

ABER: Der Compiler kann die Konstante **weg optimieren** und in den Assemblercode als **Direktooperand** eintragen, wenn diese sonst nirgends verwendet wird. => Es ist sehr von der Situation u. vom Compiler abhängig, was genau mit der Konstanten passiert!

In der **Microcontroller-Programmierung** ist die Verwendung von **#define** deshalb weiterhin sehr beliebt und verbreitet!