

Operating Systems

Assignment 1.1

Henri Heyden, Nike Pulow

stu240825, stu239549

1)

C was designed in 1972 by Dennis Ritchie to implement a portable UNIX-System / program.

2a)

GCC stands for **G**nu **C**ompiler **C**ollection.

2b)

GCC also supports C++, Objective-C, D (a dialect of C++), Fortran, Ada and Go.

3a)

In C a comparison results in a value of an int type.

3b)

In Java a comparison results in a value of a boolean type.

3c)

Nope, the bool-type was introduced in C99, but it is not a standalone type, since it has to be imported from "`stdbool.h`"

4)

Bit fields can be used to implement efficient type structures that have to run fast and not use more space than needed.

For example if one would want to implement a database system that stores student IDs, one may assign certain bits to certain semantical or syntactical rules, so that other systems like a card reader or people could interact with the structure because there is an order of the collection.

In the student example it could be that if a value of this structure would get send by cable to another system, then arranging the different sub-types in some way may influence the speed of the receiving system since there may not have to be read everything by the receiver.

5)

The intended purpose of unions is to save memory by using the allocated space for different tasks.

In some way this is a similar thought to manual garbage collection, but for parallel running tasks repeatedly using the same resources.

6)

The prefix variant changes the object and returns it whereas the postfix variant only changes the object.

7a)

Passing variables is per-se not possible in C, instead one passes a pointer to a variable if one wants to change it, since if one were to pass the variable directly as a not pointer type, then only the value of the object is transferred, so it can't modify the outer variable.

7b)

This means that by passing a pointer to the wanted object, we can access it. If one were to change the pointer itself, then there has to be a pointer to it get delivered.

7c)

It means that for example that every time you want to access something that is not in your current method, then you have to create a pointer to it, including pointers itself. With that, there is some sort of chicken-egg problem, because one can't modify a pointer without creating a pointer to it, if it can't be accessed directly.

Also, there comes the thought, why the compiler should even allow passing not constant variables without wanting to use a pointer, but there are situations, where this is still useful, like for example when you have two methods: One working with the value of the variable and one working with the variable itself.