Exercise 1

1. Cross-platform development without ĥaving multiple OSes installed natively or using several computers. Different memory and processor requirements.

2. ALU instructions are not sensitive, MMU and IO-MMU instructions are sensitive

3. Instructions that cause a trap

4. Yes, as a hypervisor can create sub-partitions

5. E.g. if the clear interrupts instruction is executed from the guest, the hosts only changes an internal data structure to mask the interrupts for the specific guest instead of executing the expensive cli instruction

6. Multiple OS could access the same frame

7. Doesn't recognize page access patterns for guest OS → often swaps potentially used page → many page faults

8. Force the guest to swap/page out some pages

Exercise 2

1. A pointer to an integer, where the address that the pointer holds cannot be changed

2. A pointer to an integer

3. A pointer to an array of 3 integers

4. A function returning a float named f1, taking a pointer to a constant integer as arg

5. Sie kann sinnvoll sein, wenn es wichtig ist, dass die Addresse an die der Pointer zeigt sich nicht ändert.

6. The function f is called with a static string as input. This will throw a segmentation fault as the char pointer that is provided to f is refering to a read-only section which cannot be written to.

7. `int* func(void);`

8. `int* (*fn_ptr)(void);`

Exercise 3