Exercise 1

1. Not frequently used, Aging, not recently used
2. Address translation needs to be fast and if the virtual address space is large, the page table will be large. Local vs. global page allocation, locking pages
3. It has to make a large number of references to a small number of pages.
4. Safety (control over virtualized resources, if one VM crashes, the others continue to run, isolation of VMs, i.e. if one is infected, it does not spread), Fidelity (identical bahaviour of running in VM or bare metal)
5. The transition between an operating system and the hypervisor
6. Virtualization abstracts hardware and isolates operating systems while containers abstract operating systems and isolate processes
7. Binary translation replaces instructions of the guest OS on the fly by the according ones of the host or hypervisor. Hardware based approaches result in lots of traps whicgh thrash the CPU caches, branch prediction tables and TLB entries. Thus binary translation may be faster as it simply replaces critical instructions with ones that emulate the very same.

Exercise 2

1. Instruktion auf zwei Pages verteilt, erster und zweiter Operand jeweils über pages
2. $\frac{2^{48}}{2^{13}} = 2^{35} Pages$
3. Der Programmcode nimmt exakt die Hälfte des Addressraums ein und ist perfekt aligned mit der Page size (8 Pages ·4 KiB/Page = 32 KiB. Die Daten umfassen 2 Byte mehr als mit 4 Pages speicherbar wären. Die Daten bennötigen also 5 Pages. Der Stack bennötigt 4 pages, was insgesamt zu 17 Pages führt. Da aber nur 16 verfügbar sind passt der Prozess nicht in den Addressraum
4. Mit 512B Page größe, stehen insgesamt 128 Pages zur Verfügung. 64 davon bennötigt der Programmcode, 33 Pages werden für die Daten bennötigt und 31 für den Stack. Damit passt der Prozess in den Addressraum

Exercise 3

1. 2
2. 0
3. 1
4. 0

Exercise 4

1. 1, Prints the first value of the first subarray
2. 1, prints the starting address of the array which is also the staring address of the first subarray which is the first element of the first subarray
3. 2, adds one integer step to the address of the first element of the first subarray
4. 3, adds one step to the address of the outer array, i.e. gets the address of the first element of the second subarray
5. as above
6. the innermost addition adds two subarray steps to the address of the array, i.e. this is the address of the third subarray. by subtracting one, we arrive at the second element of the second subarray, i.e. 4 is printed

Exercise 5

1. $\frac{s}{p} \cdot e + p/2$ Pages per process on average each having an entry size of e
2. The smaller the page size, the smaller the second term, but the higher the first one. The otherway round, the larger the page size, the less entries, but the more thrashing
3.
4.