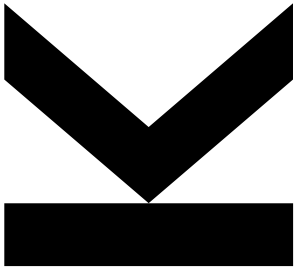
Memory allocation



Systems Programming

JOHANNES KEPLER
UNIVERSITÄT LINZ

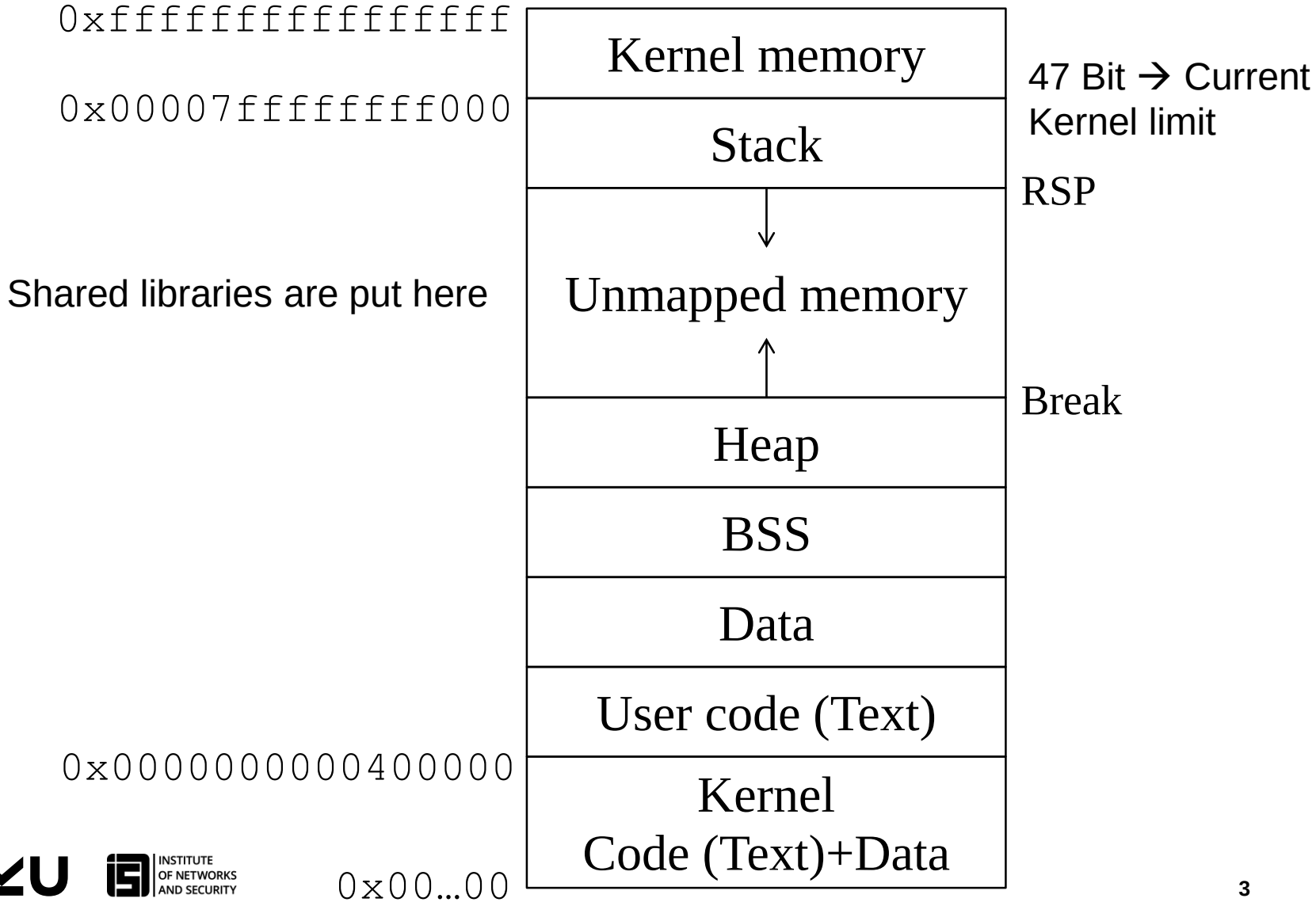
Michael Sonntag
Institute of Networks and Security

INSTITUTE
OF NETWORKS
AND SECURITY

Memory organization

- **Physical memory addresses**
  - Memory addresses referring to RAM chips (“real”)
  - Exist always in full size (as far as chips are present)
  - Only visible to OS – we will never see it in our programs!
- **Virtual memory addresses**
  - Memory addresses used by programs (“logical”, “virtual”)
    - What we use here!
  - Mapping to physical memory addresses
    - Support by OS together with CPU’s Memory Management Unit (MMU)
  - Mapping to disk space: Swap space/partition
  - Flexibility for programmers
    - Do not care about how much memory is physically available
    - Do not care about how virtual addresses are mapped to physical ones
  - “Exists” only as far as actually used
    - No mapping to physical addresses if not reserved; access causes fault

Linux memory layout



Memory – factorial-main.s

- Start program in debugger and set breakpoint at start
- Find process ID through **`ps -A`**
- `cat /proc/25364/maps`

Code →
Data →

```
00400000-00401000 r-xp 00000000 00:25 514                /mnt/factorial-main
00600000-00601000 r-xp 00000000 00:25 514                /mnt/factorial-main
00601000-00602000 rwxp 00001000 00:25 514                /mnt/factorial-main
7ffff7bda000-7ffff7bdb000 r-xp 00000000 00:25 512        /mnt/libfactorial.so
7ffff7bdb000-7ffff7dda000 ---p 00001000 00:25 512        /mnt/libfactorial.so
7ffff7dda000-7ffff7ddb000 r-xp 00000000 00:25 512        /mnt/libfactorial.so
7ffff7ddb000-7ffff7ddc000 rwxp 00001000 00:25 512        /mnt/libfactorial.so
7ffff7ddc000-7ffff7dfc000 r-xp 00000000 fd:00 37071      /usr/lib64/ld-2.17.so
7ffff7ff6000-7ffff7ffa000 rwxp 00000000 00:00 0
7ffff7ffa000-7ffff7ffc000 r-xp 00000000 00:00 0          [vdso]
7ffff7ffc000-7ffff7ffe000 rwxp 00020000 fd:00 37071      /usr/lib64/ld-2.17.so
7ffff7ffe000-7ffff7fff000 rwxp 00000000 00:00 0
7ffffffde000-7ffffffff000 rwxp 00000000 00:00 0          [stack]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0  [vsyscall]
```

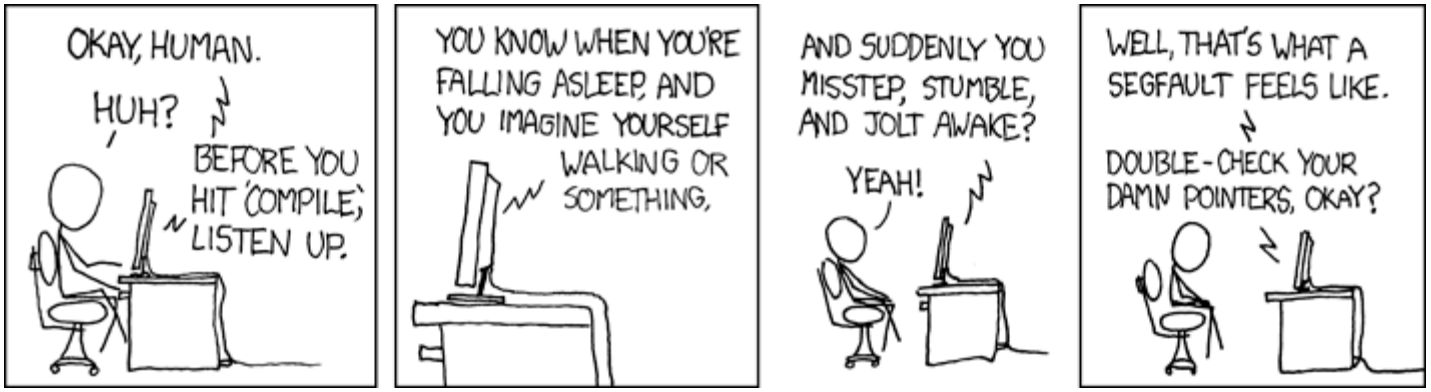
Linux memory layout – X86-64

- **Code**: Instructions of the program
  - Read-only
- **Data**: Data of the program (=initialized)
  - Read-Write
- **BSS**: Buffers of the program (=uninitialized)
  - Read-Write
- **Heap**: Dynamically allocated memory
  - Read-Write
- **Stack**: Temporary data, procedures
  - Read-Write
- **Unmapped memory**: Memory not mapped to physical addresses
  - Access leads to segmentation fault
- **Break**: First non-usable (mapped) memory address

Dynamic memory allocation

- **Grow/Shrink Mapped Address Space**
  - **`brk`** system call
  - RAX contains **`12`** (system call number of **`brk`**)
  - RDI contains requested break
  - **`brk`** returns the new break in RAX or zero, if there is not enough physical memory or swap space
    - Actual new break might be larger than requested (Linux “might”=will round up to the nearest page, typ. 4 kB)
- **Problem**
  - Increment break for space of new object 1
  - Increment break for space of new object 2
  - What if object 1 is no longer needed?
    - Gap of mapped memory addresses that are not used anymore
- **Solution**
  - Memory manager keeping track of used memory; reuses gaps

Be careful where you point to...



XKCD, Compiler Complaint, https://xkcd.com/371/

JOHANNES KEPLER
UNIVERSITÄT LINZ

INSTITUTE
OF NETWORKS
AND SECURITY

https://www.ins.jku.at

THANK YOU FOR YOUR ATTENTION!

Michael Sonntag
michael.sonntag@ins.jku.at
+43 (732) 2468 - 4137
S3 235 (Science park 3, 2nd floor)

JOHANNES KEPLER
UNIVERSITÄT LINZ
Altenberger Straße 69
4040 Linz, Österreich
www.jku.at