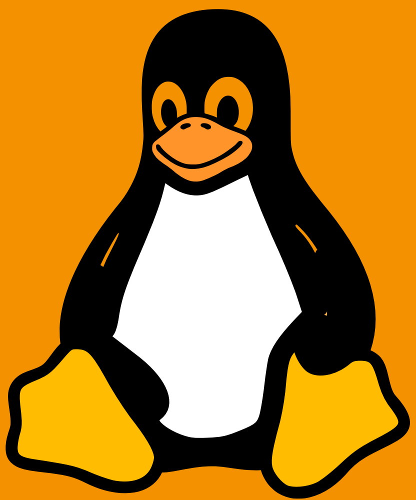
Technische Hochschule Rosenheim
Technical University of Applied Sciences

Prof. Florian Künzner
Technical University of Applied Sciences Rosenheim, Computer Science

OS 12 – Memory Management

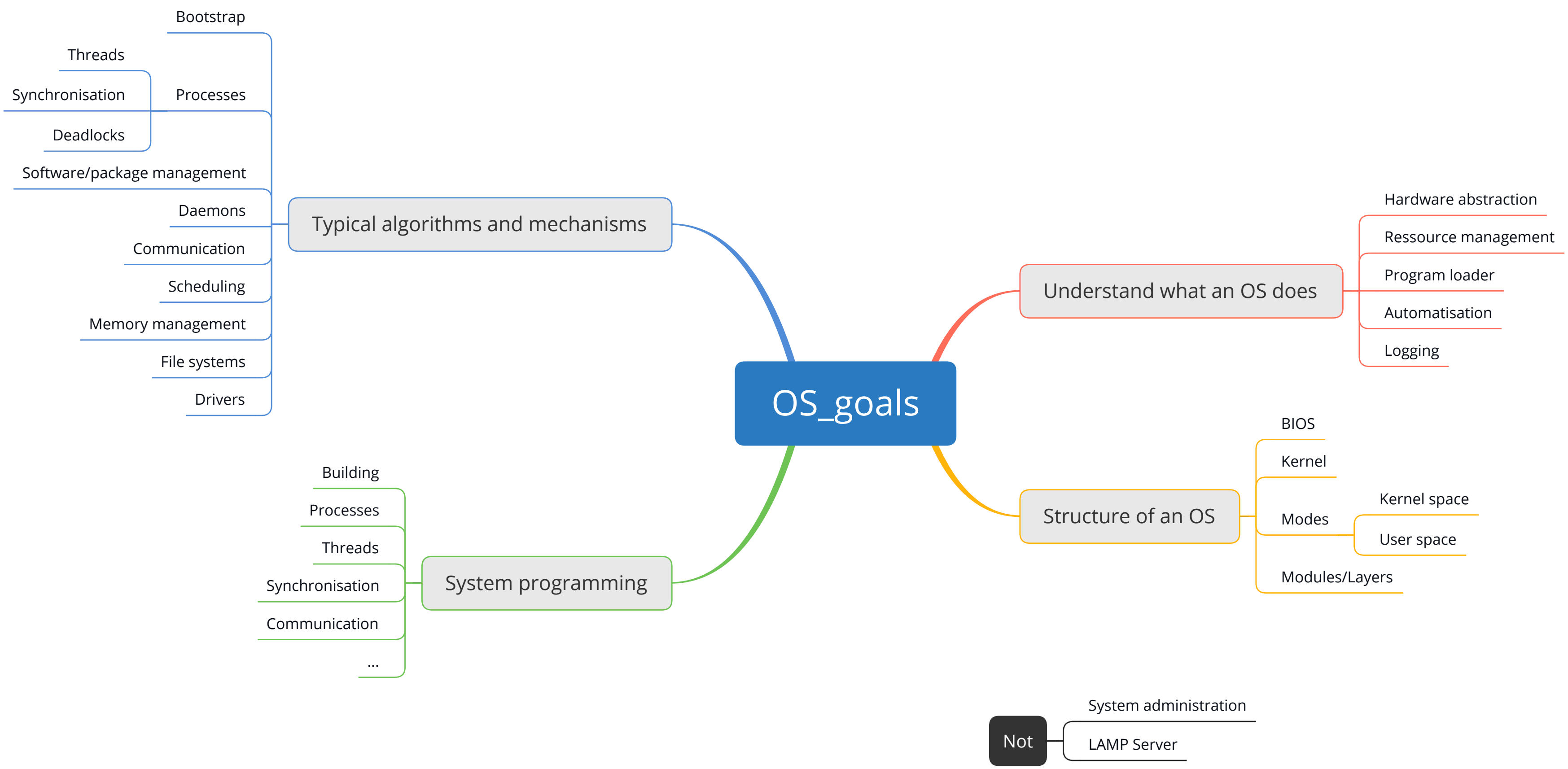

source: iconspng.com

The lecture is based on the work and the documents of Prof. Dr. Ludwig Frank

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal

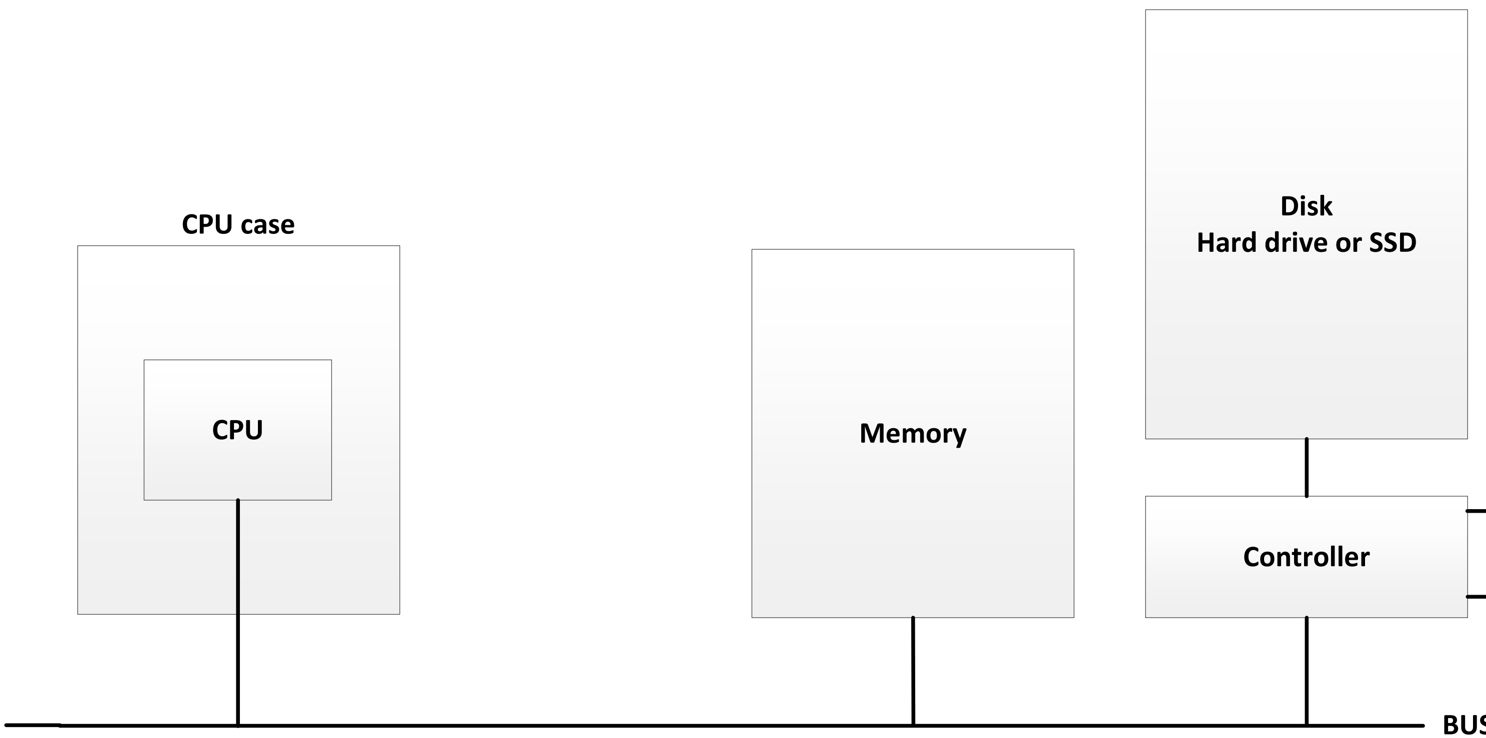
OS::Memory Management

- Caching
- Partitioning
- Fragmentation
- Allocation strategies
- Swapping
- Virtual memory

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Memory overview



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Memory management requirements

Requirement	Description
Retrievability	Who is the **owner** of the **memory area**?
Protection	**Protect the memory areas** of the different processes against unwanted interference (read/write).
Efficient usage	**Memory** is (always) too **small**.
Sharing	Share memory between processes (**shared memory**).
Logical organization	A process wants to see its memory as a **sequence of bytes**.
Transparency	The **programmer doesn't now** at programming time **how much memory** on the target computer **exists**. The memory management should transparently manage this.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Intro

Towards modern memory management...

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Caching

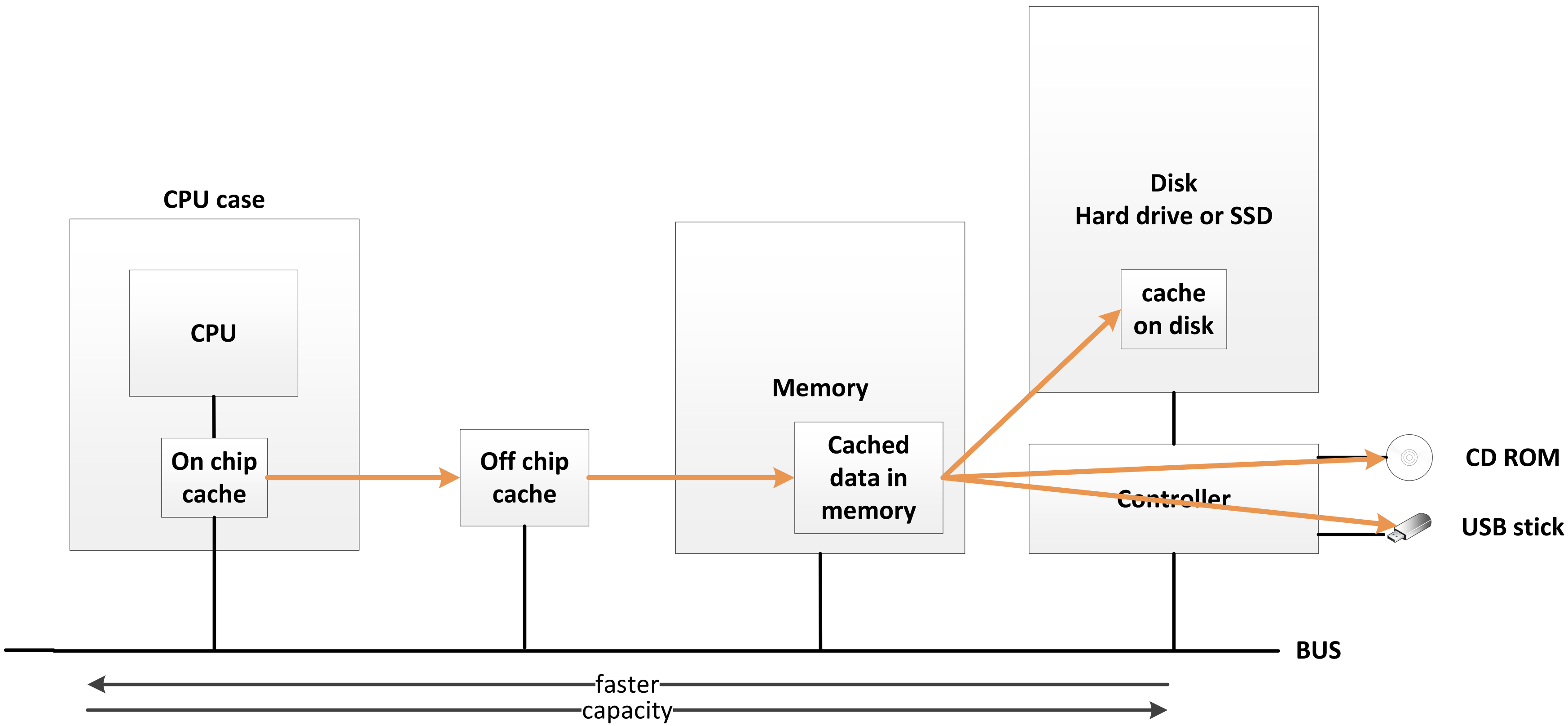
Problem

The **CPU is much faster** than the memory and all the other periphery (e.g. network).

CAMPUS Rosenheim
Computer Science

Caching

Idea Introduce caches (data buffers) on various places.



CAMPUS Rosenheim
Computer Science

Caching terms

Hit

A hit occurs on repeated access of the same memory address.

Fault

Occurs on the first access. The cache is too small to buffer all data.

Locality

- **temporal**: If a memory address is accessed, there is a high probability that it will be accessed again very soon.
- **spatial**: If a memory address is accessed, it is very likely that surrounding addresses will be accessed, too.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Caching examples

On chip cache

A small cache inside the CPU: L1, L2, and L3. The average access time appears faster.

Off chip cache

A cache outside of the CPU (e.g. a special PCIe device).

Disk cache

A cache inside the disk or inside the OS. Speeds up the access time to the data on disk.

Internet cache

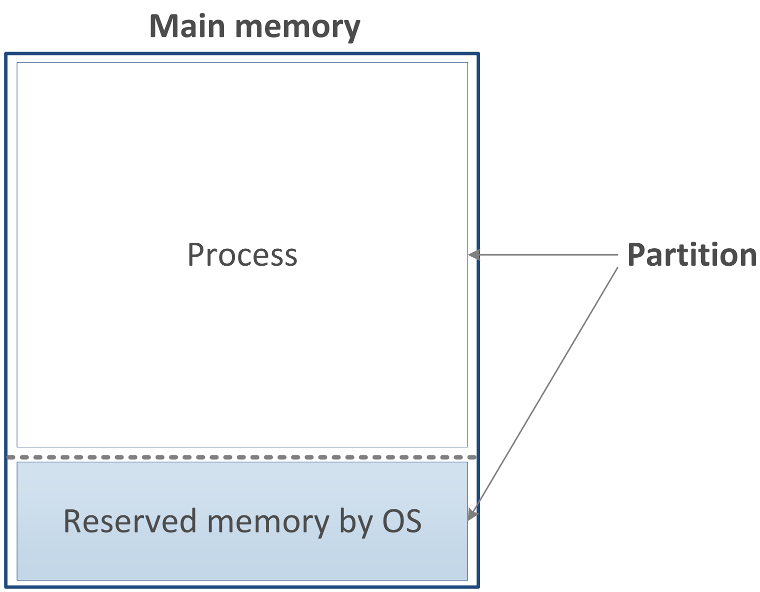
A internet proxy (e.g. squid) placed in the local network. If everyone accesses the same website or downloads the same file.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

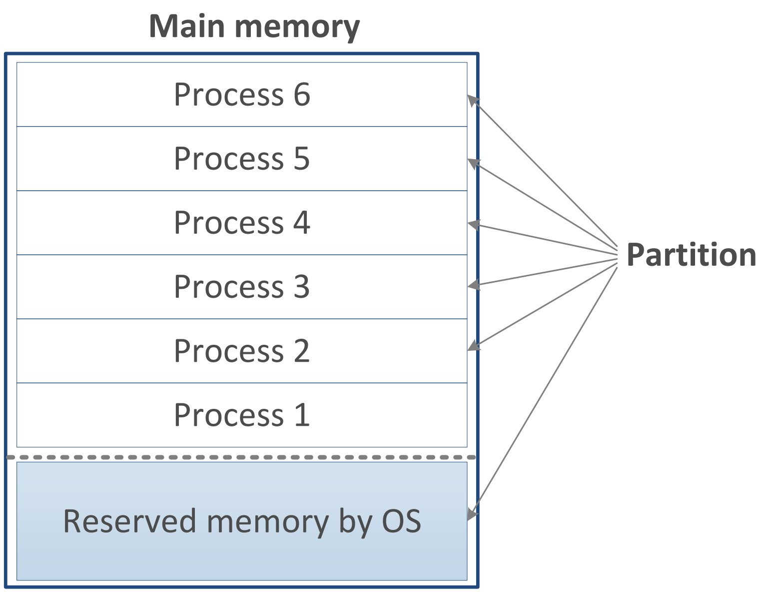
Partitioning

Single process mode



The main memory has to be divided into the OS part and the process part. If the process needs more memory that is physically available it can't run.

Multi process mode



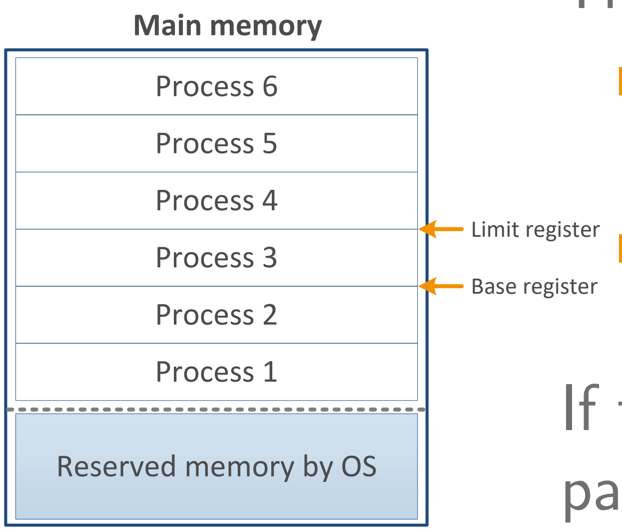
The main memory is divided into the OS part and the rest is partitioned between the processes.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Problem 1: memory protection

How to detect that a process accesses memory that it does not own?



The hardware offers two additional registers:

- **Base register**: start address of memory partition
- **Limit register**: end address of memory partition

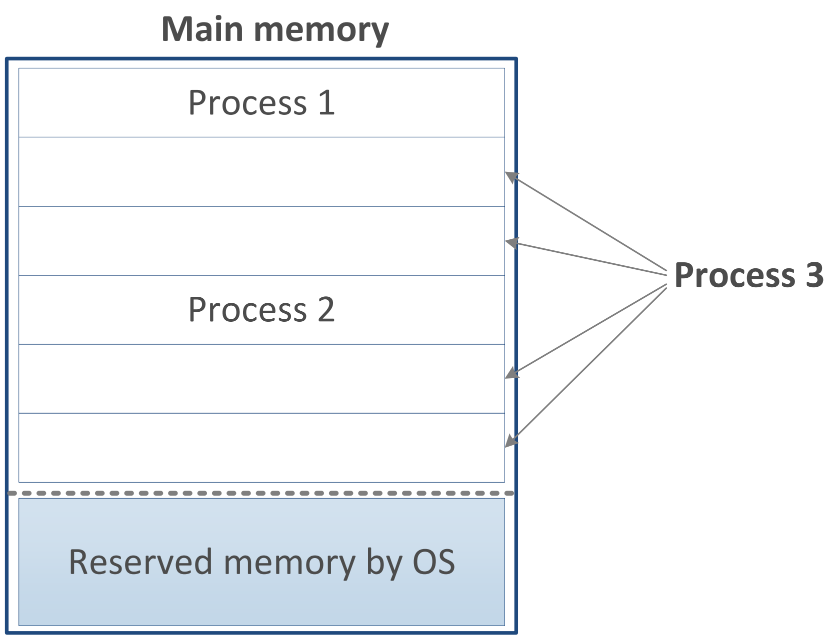
If the process access an address outside the partition it is interrupted: the OS does than the error handling.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Problem 2: position dependent code

Where should the program be loaded? What happens with the addresses inside the process?

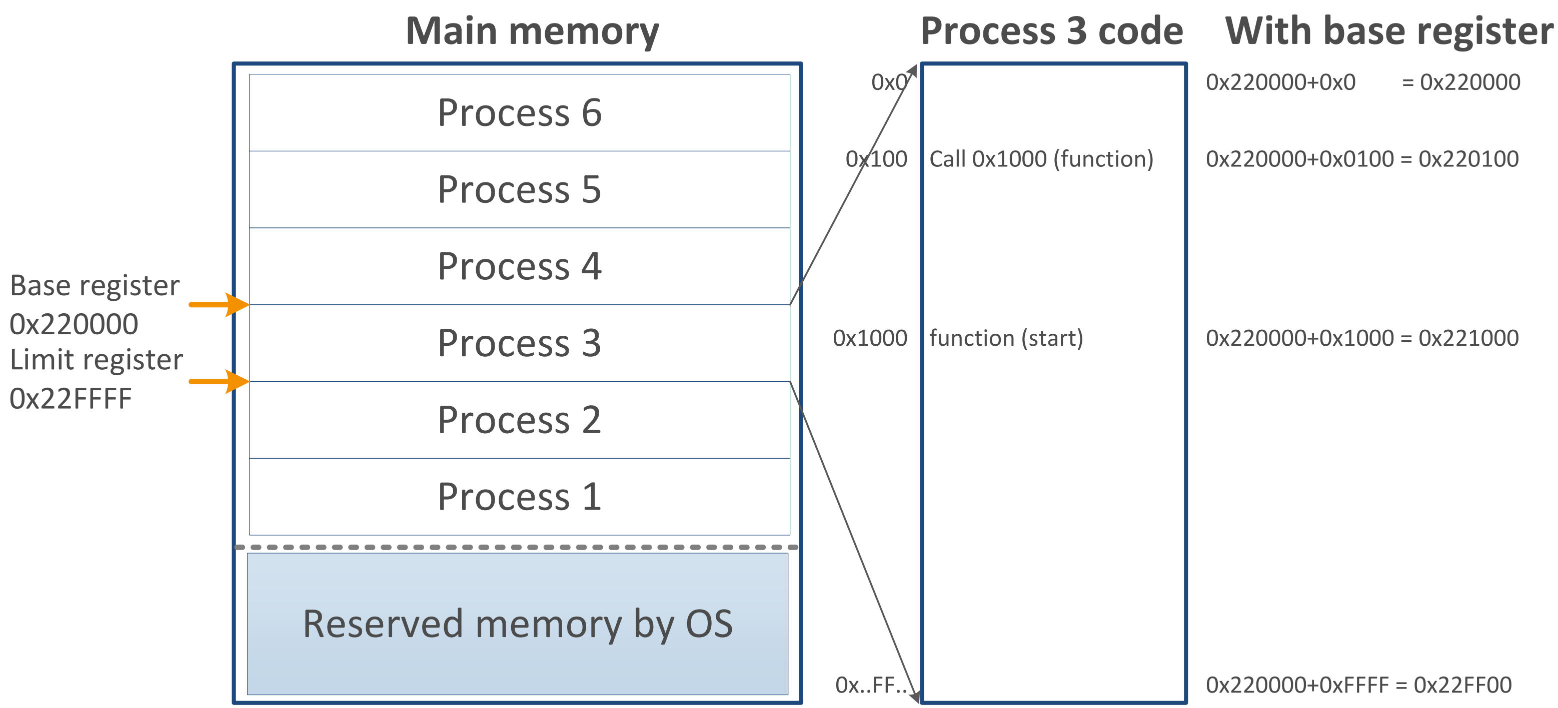


CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Problem 2: position independent code

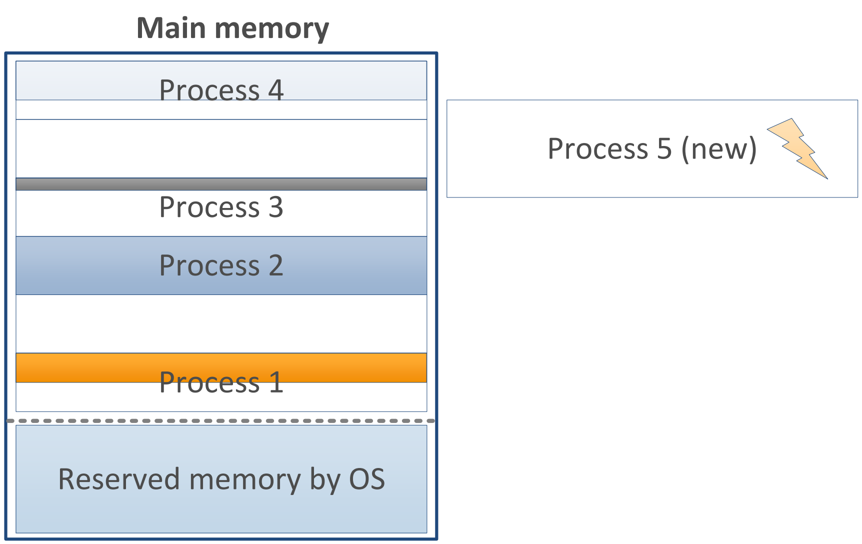
Build the program as it would start at address 0. On every memory access, the base register is added to the address.



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Fragmentation



Internal fragmentation

Partitions are allocated by processes, but **not fully used**. May be solved by variable partition size.

External fragmentation

A lot of free partitions can't be used, because they are **too small** for some processes. May be solved by dynamically move the partitions (base and limit register required). But this will take some time.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

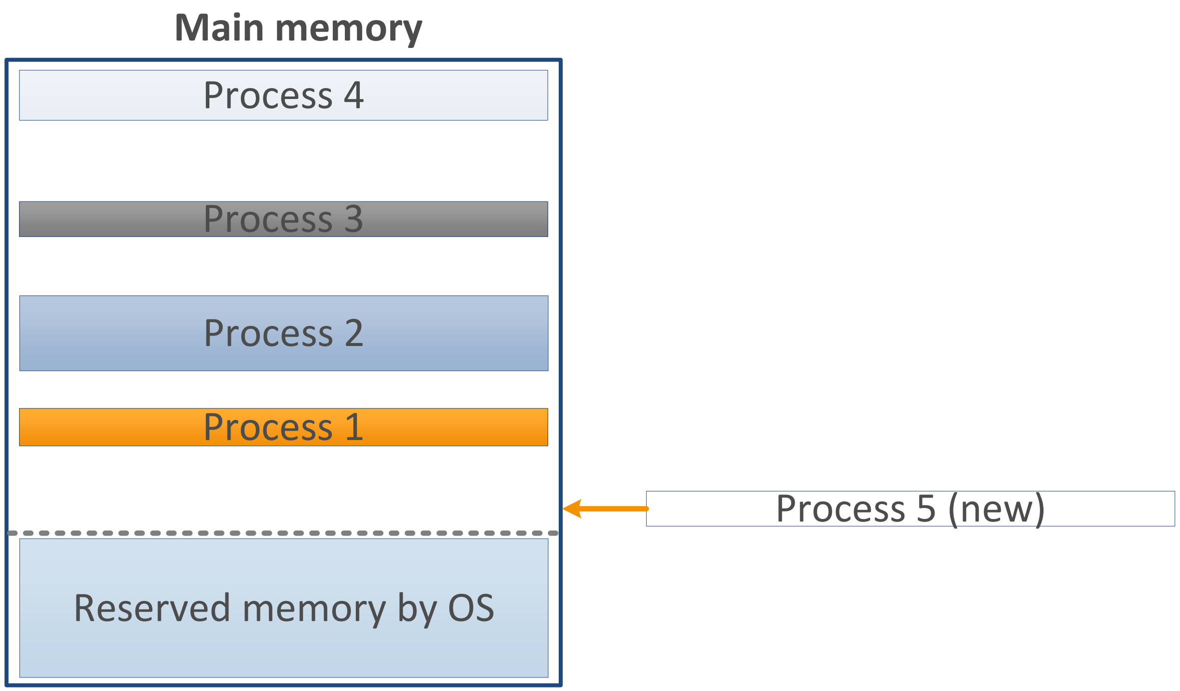
Allocation strategies

How to find the best place for the new partitions in the memory?

CAMPUS Rosenheim
Computer Science

First fit

Starts at the beginning of the main memory and takes the first memory area that fits.

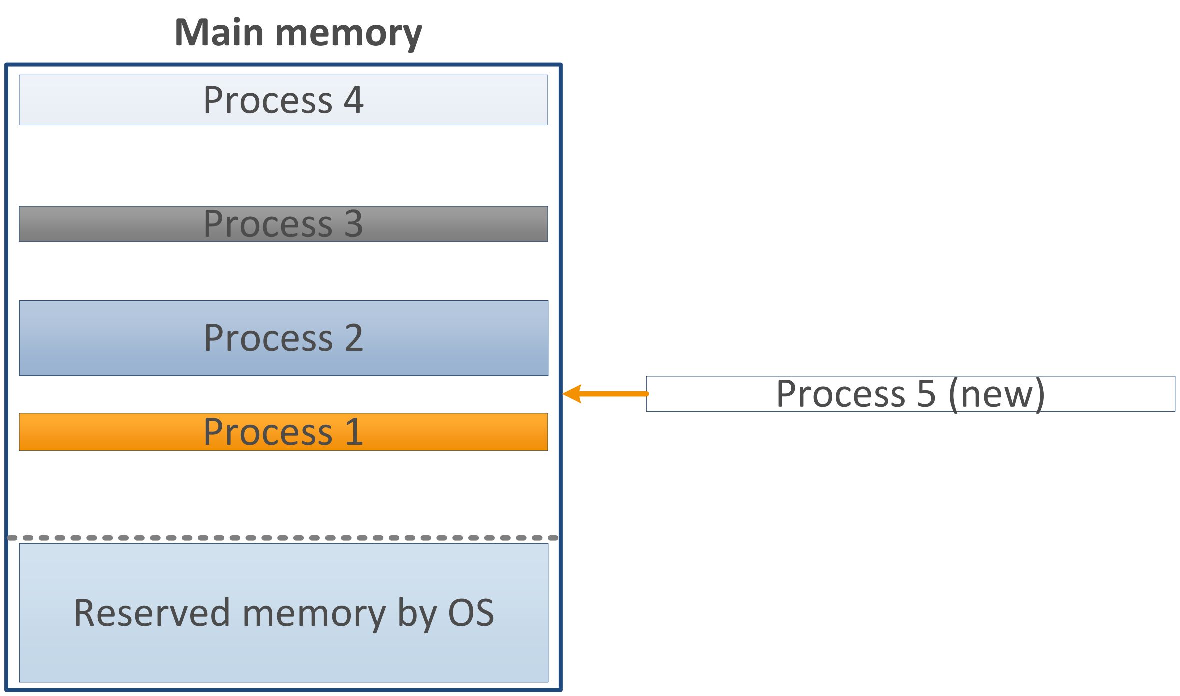


CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Best fit

Looks into all free memory areas and takes the memory area that fits best.

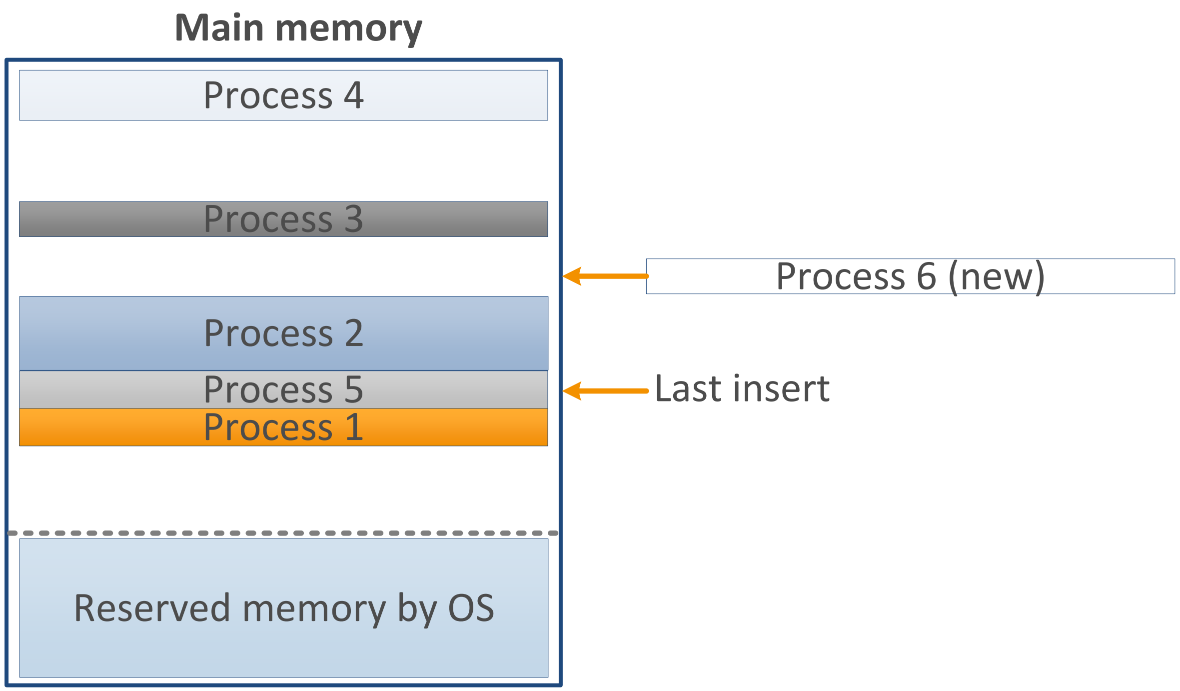


CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Next fit

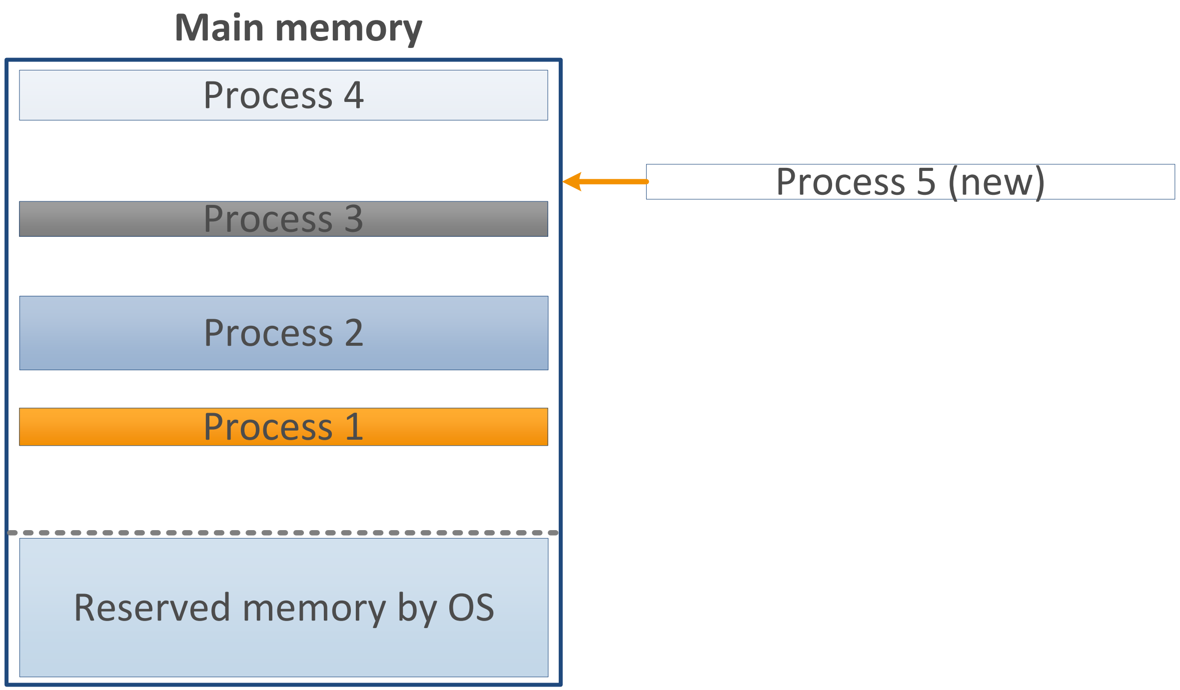
The same as first fit, but it starts at the point, where the last search ended.



CAMPUS Rosenheim
Computer Science

Worst fit

Looks into all free memory areas and takes the memory area that fits worst.



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Allocation strategies summary

Simulations show that generally best fit is worse than first and next fit.

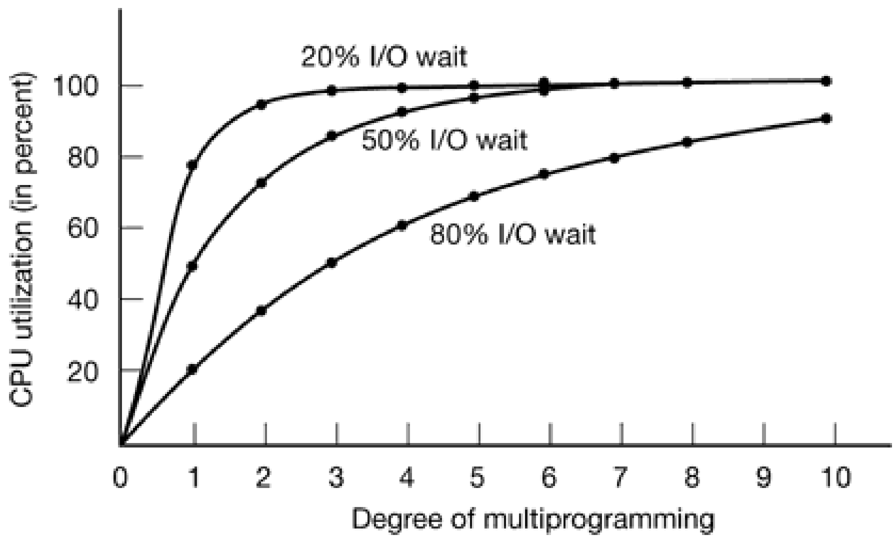
Worst fit is the worst.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Still existing problems

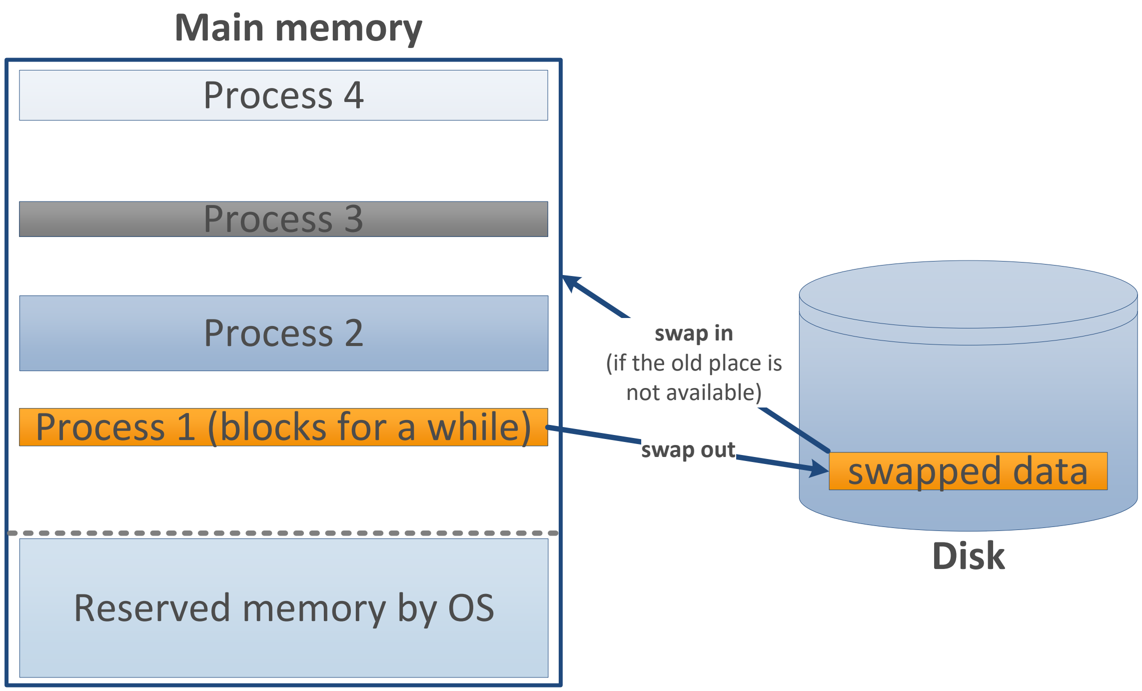
- The number of processes is limited by the number of partitions.
- The processes often don't use the memory allocated to them for a long time.
- A lot of processes are required to fully use the CPU.



Source: Modern operating systems (Tanenbaum, second edition)

Swapping idea

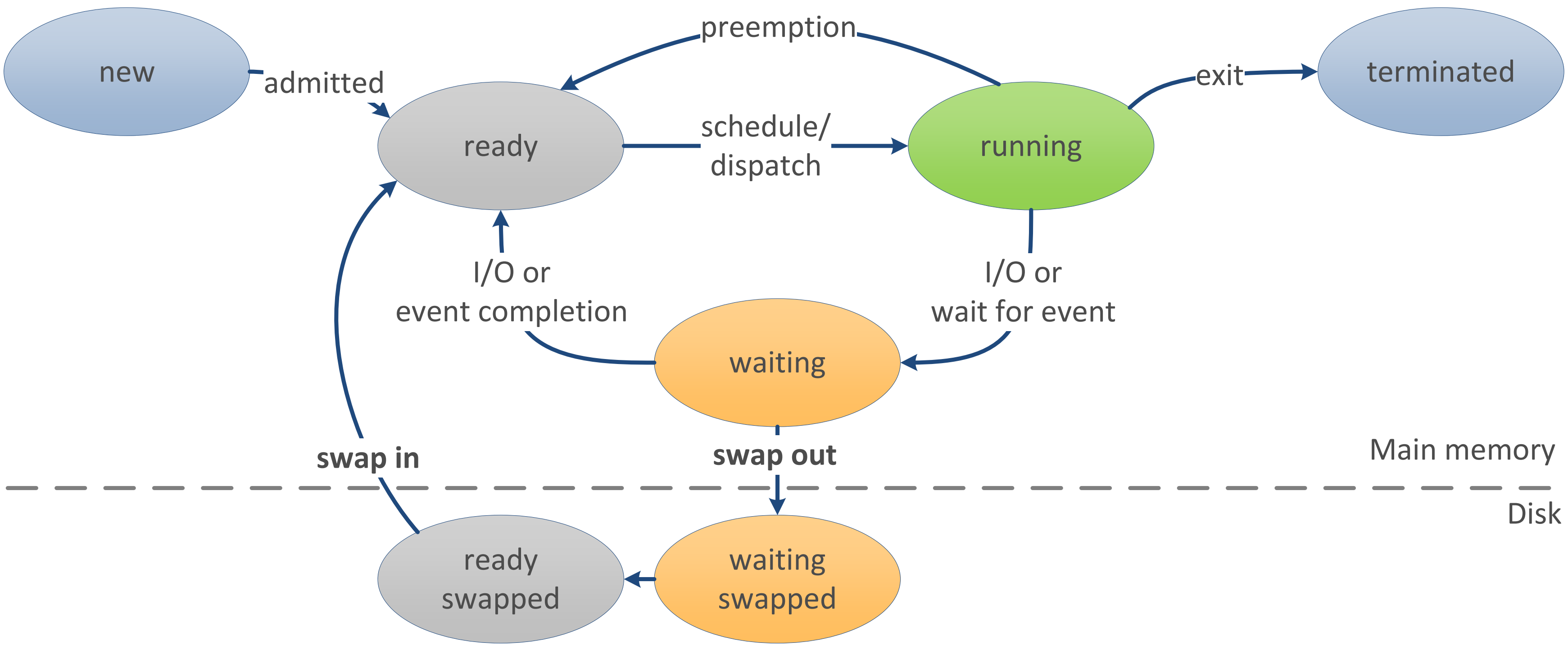
Swap the memory of a process that is not actively running to the disk.



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Enhanced process states



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Where are we?

Achieved so far

The sum of all processes memory can have more memory than is physically available.

Still existing problems

- Every process is limited by the physically available memory size.
- The processes often don't use the memory allocated to them for a long time.
- Problem with fragmentation still exists (dynamically move of partitions takes a long time).
- Swapping is time consuming. The advantage is often difficult to evaluate.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Virtual memory

Idea

Only load the required memory parts of a process into the memory.

Requirement

But neither the user nor the programmer should notice anything about it.

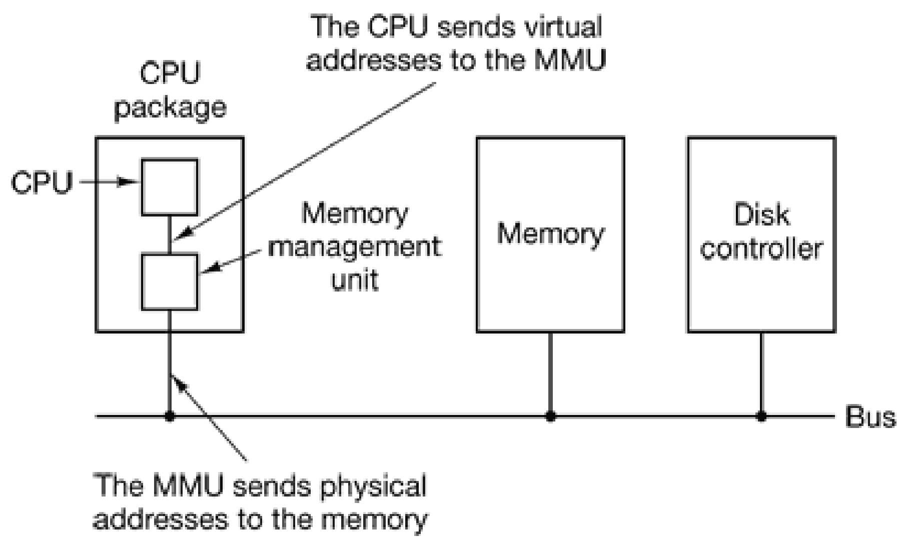
=> Virtual memory with virtual addresses!

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Virtual address

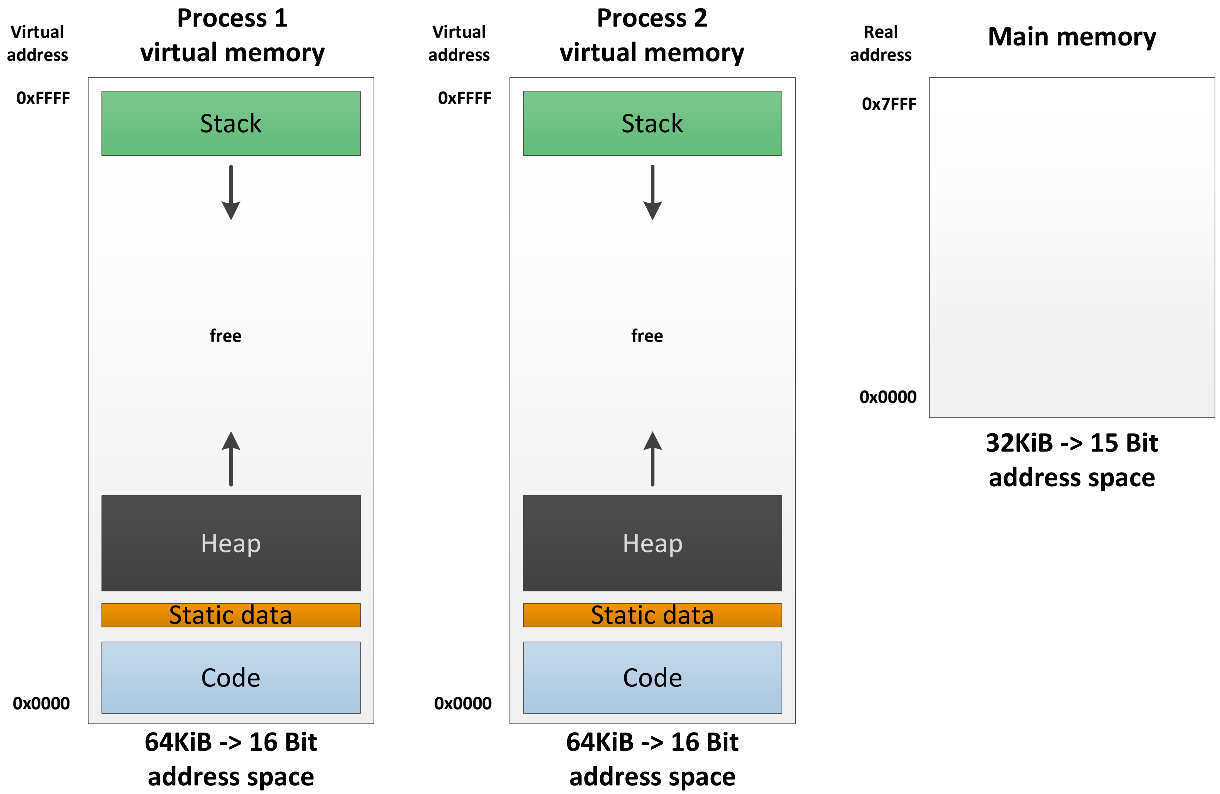
The MMU (a special hardware inside the CPU) supports the address calculation: virtual address -> real address.



Source: Modern operating systems (Tanenbaum, second edition)

CAMPUS Rosenheim
Computer Science

Virtual address space



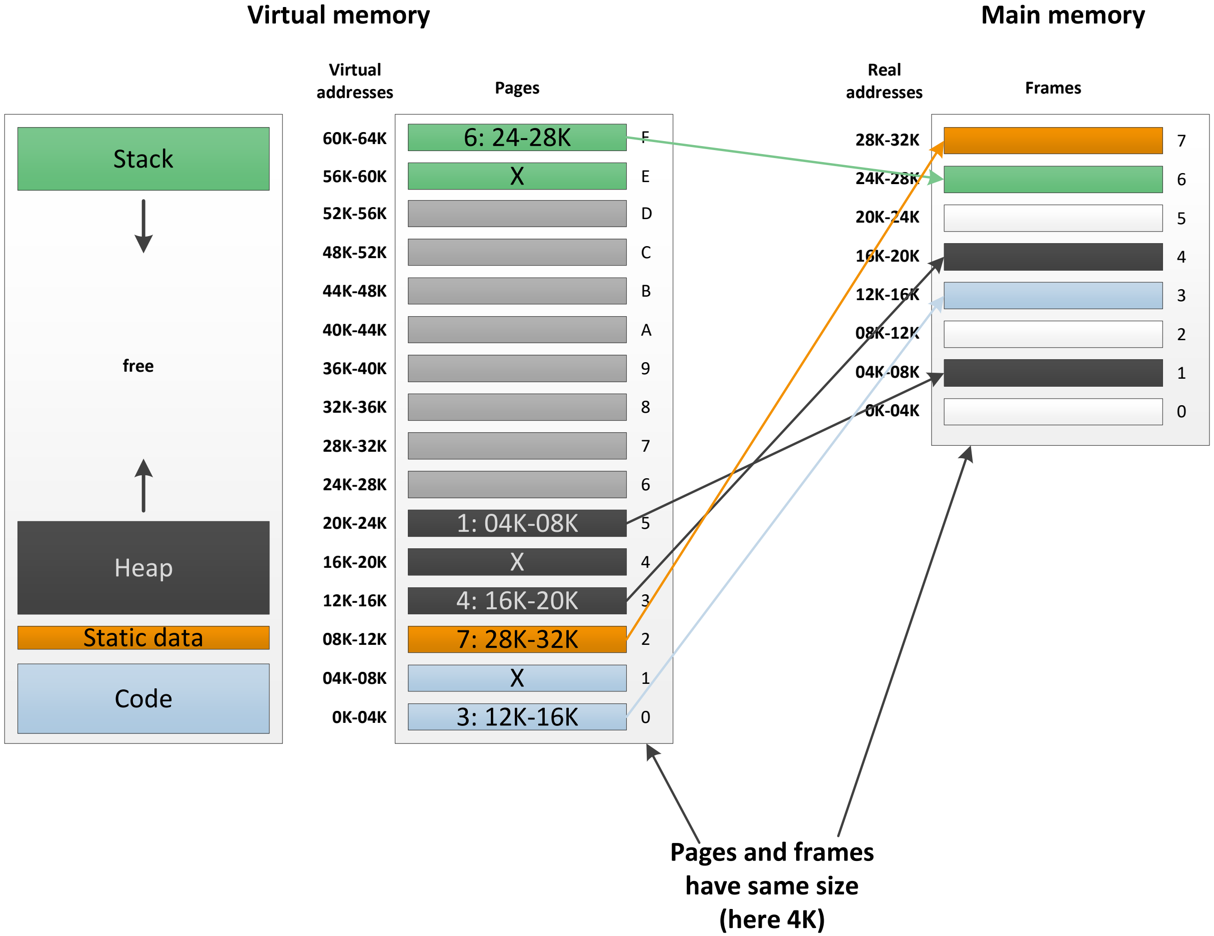
Virtual address space
Each process has its own virtual address space. The theoretically available memory.

Real address space
Physically available address space

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

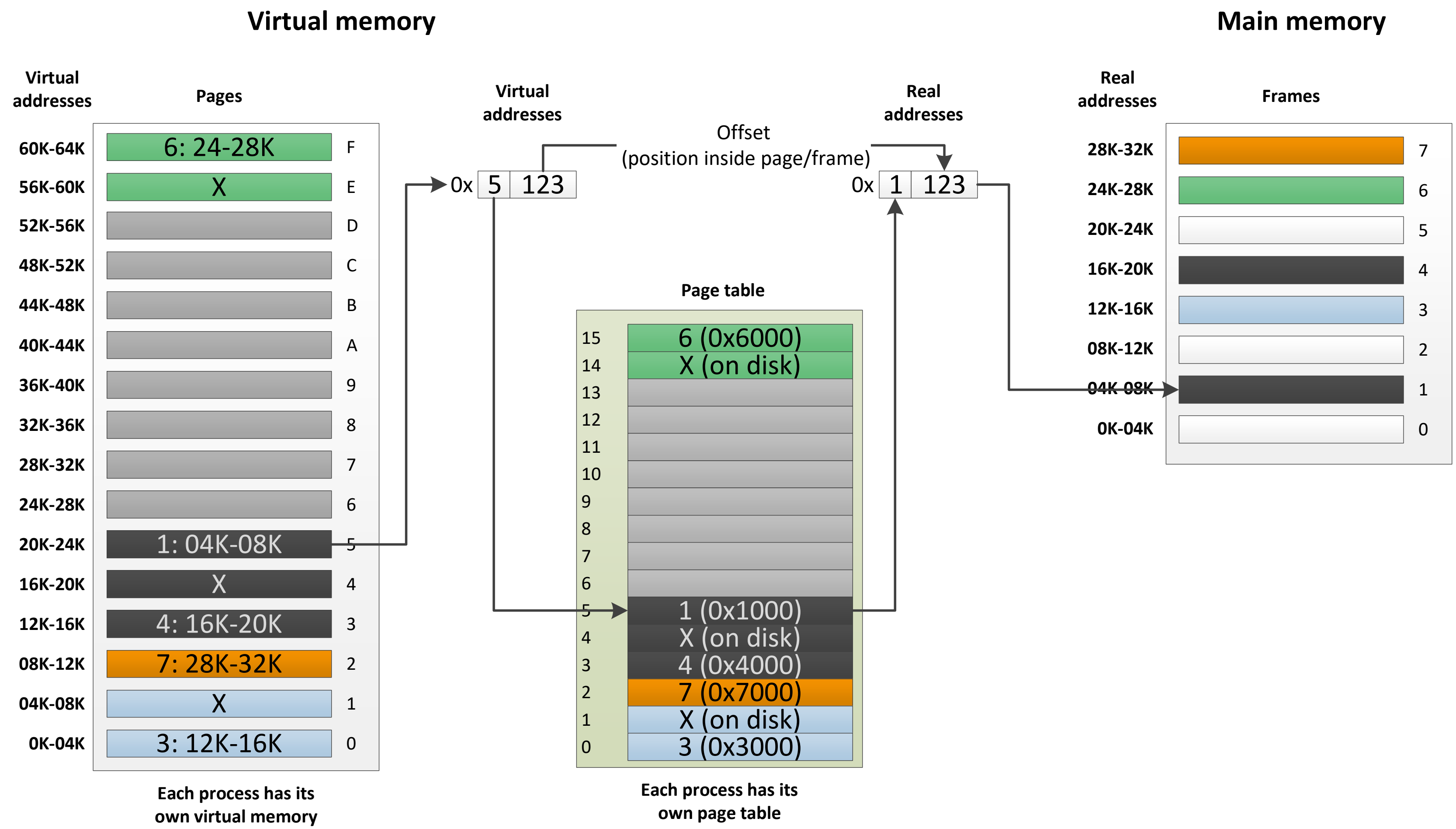
Pages and frames



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

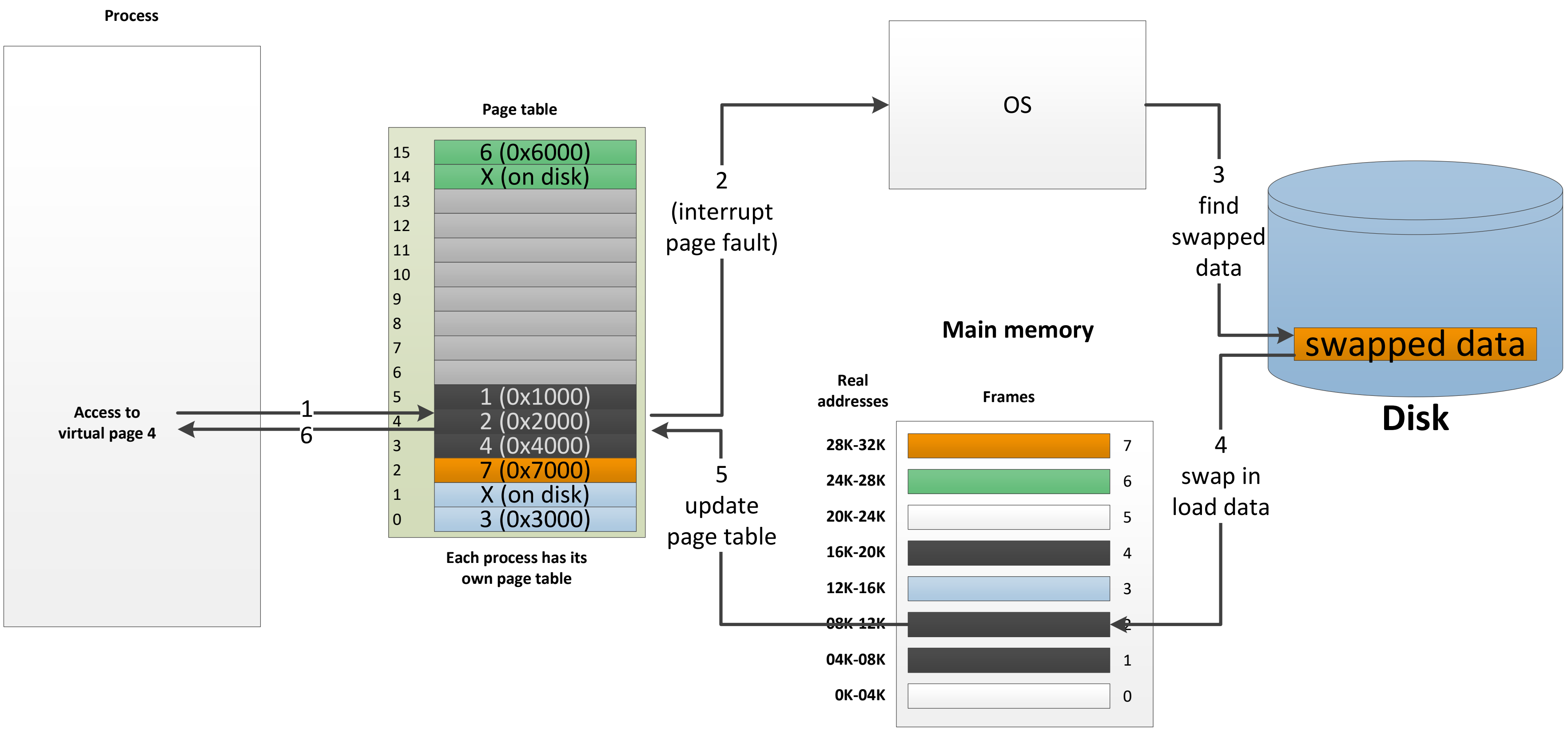
Page table



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Page fault

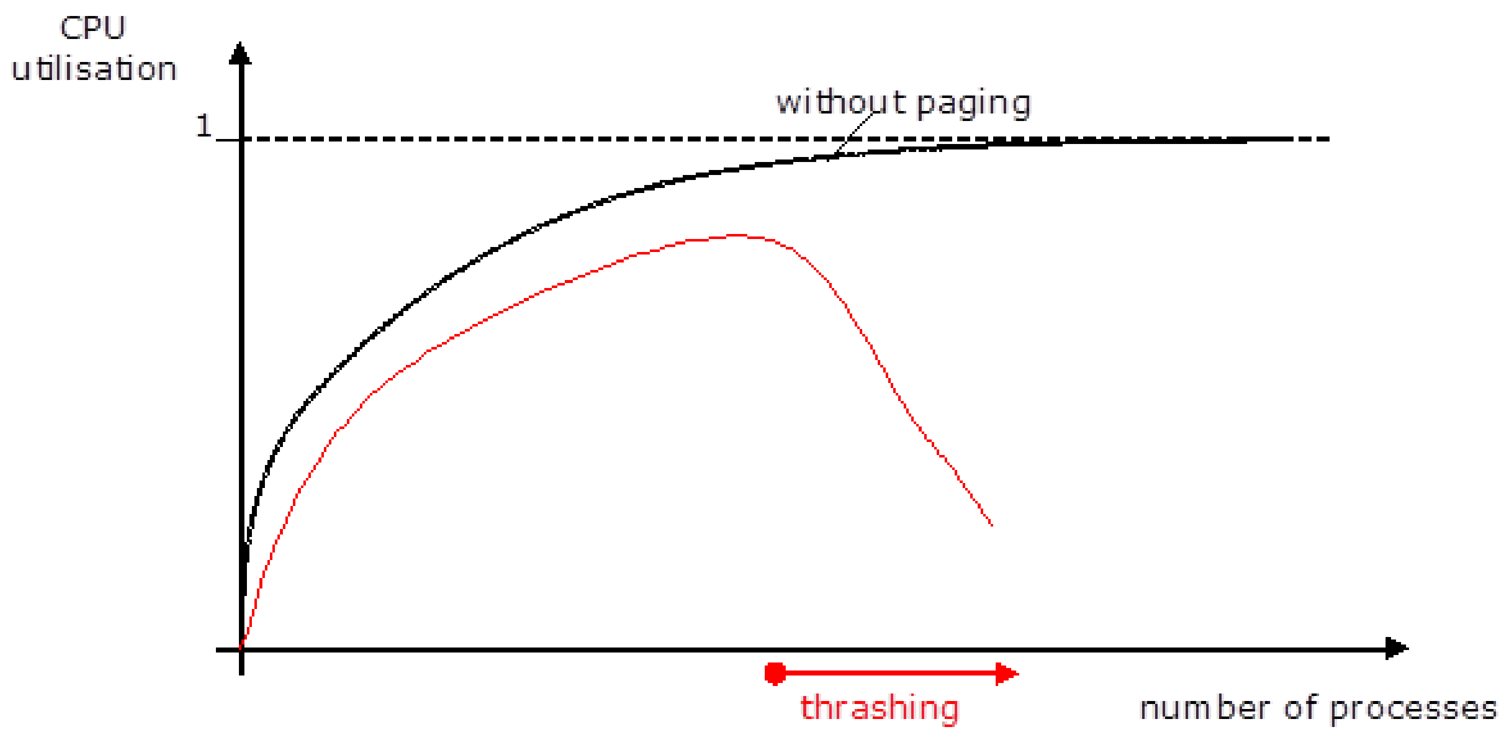


CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Thrashing

Thrashing happens if processes with swapped pages are concurrently be activated and the OS needs swap in and swap out very often.



CAMPUS Rosenheim
Computer Science

Summary and outlook

Summary

- Caching
- Partitioning
- Fragmentation
- Allocation strategies
- Swapping
- Virtual memory

Outlook

- Shared libraries
- User management
- File systems