Prof. Dr. Carsten Vogt
Exercises „Operating Systems and Distributed Systems 2“
2019/20

Technology
Arts Sciences
TH Köln

Virtual Memory – Solution

Exercise 1: Segmented Virtual Memory

- Given:
  - A (very small) main memory with a size of 1000 bytes
  - A segment table:

segment number	physical start address	length	access rights
1	0	150	read/write
2	disk block 12	40	write only
3	150	180	read only
4	430	200	read/write
5	disk block 21	110	read/write
6	830	120	read/write

- The table below shows a sequence of segment accesses with their virtual addresses. Calculate the corresponding physical addresses (i.e. byte numbers in main memory) – even if the access is not permitted.
  Moreover, state whether the access is permitted – and, if not permitted, why not.
  If the segment is currently stored on hard disk, state this in the rightmost column of the table. Load the segment into main memory (using the "best-fit strategy"), modify the corresponding entry of the segment table and calculate the physical main memory address that results after this operation.

operation with virtual address	physical byte address	access permitted?
read segment 4, offset 0	***430***	***yes***
write segment 6, offset 90	***920***	***yes***
write segment 3, offset 78	***(228)***	***no (read only)***
write segment 2, offset 35	***985***	***load segment before***
read segment 1, offset 155	***(155)***	***no (offset exceeds length)***
read segment 5, offset 80	***710***	***load segment before***

- **How to find out where to load the segments to:**
  - Determine the free spaces in main memory by using the physical start addresses and lengths from the segment table:
    (start: 330, length: 100), (start: 630, length: 200), (start: 950, length: 50)
  - Load segment 2 at addresses 950 ff. (best fit!)
  - Load segment 5 at addresses 630 ff.

Exercise 2: Paged Virtual Memory

- Given:
  - A paged virtual memory with a page size of 2 KB (= 2048 bytes).
  - A part of the page table of a process:

virtual page number	start address of physical page frame
0	14336
1	24576
2	disk block 11
3	4096
4	0
5	disk block 2
6	8192
7	...
...	...

- Do the following:
  - For the virtual addresses in the table, calculate the corresponding physical main memory addresses (if the page is currently stored in main memory) or determine the number of the disk block and the offset within the block (if the page is currently stored on hard disk).

virtual address	physical address
0	***14336***
7500	***5452***
4500	***block 11, offset 404***
2000	***16336***
11000	***block 2, offset 760***

13000	***8904***

(calculated as in the example in the lecture)

- Assume now that virtual page no. 3 is evicted to disk block 17, thus freeing a page frame in main memory. Virtual page no. 5 is then loaded into this page frame.
  - How must the page table be modified?

virtual page number	start address of physical page frame
...	...
3	**block 17**
...	...
5	**4096**
...	...

  - Where and how does, in the second table, the mapping from virtual addresses to physical addresses change?

virtual address	physical address
...	**...**
7500	***block 17, offset 1356***
...	**...**
11000	***4856***
...	**...**