Solution of Exercise Sheet 5

Exercise 1 (Memory Management)

1.	Mark memory r cur.	nanagement met	thods that cause	e internal fragmentation to oc-
		titioning		
2.	Mark memory r cur.	nanagement met	hods that cause	external fragmentation to oc-
	☐ Static partiti ☑ Dynamic par ☑ Buddy memo	titioning		
3.	Explain how ex	ternal fragmenta	tion can be fixe	ed.
	By defragmenta	ntion. For virtual	memory, extern	nal fragmentation is irrelevant.
4.		ory management ock, which fits b		searches in the entire address are request.
	☐ First Fit	\square Next Fit	\boxtimes Best fit	\square Random
5.			_	earches for the first free block inning of the address space.
	⊠ First Fit	\square Next Fit	\square Best fit	\square Random
6.		ory management the end of the a	_	agments quickly the large area
	☐ First Fit	\boxtimes Next Fit	\square Best fit	\square Random
7.	Mark the memory propriate block.		concept that se	elects randomly a free and ap-
	☐ First Fit	\square Next Fit	\square Best fit	⊠ Random
8.	Mark the memo from the latest		concept that sea	arches for a free block, starting
	☐ First Fit	⊠ Next Fit	\square Best fit	\square Random
9.	Mark the memorand is slow.	ory management	concept that p	produces many mini-fragments

Content: Topics of slide set 5 Page 1 of 11

 \square First Fit \square Next Fit \square Best fit \square Random

Exercise 2 (Buddy Memory Allocation)

The Buddy method for allocating memory to processes shall be used for a memory with a capacity of $1024\,\mathrm{kB}$. Perform the provided operations and give the occupancy state of the memory after each operation.

	1024 KB													
65 KB Anforderung => A	А	КВ												
30 KB Anforderung => B	А	B 32 64 KB	256	КВ	512	КВ								
90 KB Anforderung => C	А	B 32 64 KB	С	128 KB	512	КВ								
34 KB Anforderung => D	А	B 32 D	С	128 KB	512	КВ								
130 KB Anforderung => E	А	B 32 D	С	128 KB	E	256 KB								
Freigabe C	А	B 32 D	128 KB	128 KB	E	256 KB								
	А	B 32 D	256	КВ	Е	256 KB								
Freigabe B	А	32 32 D	256	КВ	E	256 KB								
	Α	64 KB D	256	КВ	E	256 KB								
275 KB Anforderung => F Nicht möglich, weil keine 275 kB am Stück frei	А	64 KB D	256	КВ	Е	256 KB								
145 KB Anforderung => G	А	64 KB D	(j .	Е	256 KB								
Freigabe D	А	64 KB 64 KB	(3	E	256 KB								
	Α	128 KB	(G .	Е	256 KB								
Freigabe A	128 KB	128 KB	(j .	Е	256 KB								
	256	KB	(9	E	256 KB								
Freigabe G	128 KB	128 KB	256	KB	E	256 KB								
		512	КВ		Е	256 KB								
Freigabe E		512	KB		256 KB	256 KB								
		512	КВ		512	КВ								
				102	4 KB									

Exercise 3 (Real Mode and Protected Mode)

1. Describe the functioning of the real mode.

Content: Topics of slide set 5 Page 2 of 11

Each process can access the entire memory, which can be addressed.

2. Explain why it is impossible to use real mode for multitasking operation mode.

It provides no memory protection.

3. Describe the functioning of the protected mode.

Each process can only access its own virtual memory. Virtual memory addresses translates the CPU with the MMU into physical memory addresses.

4. Describe what virtual memory is.

Each process has a separate address space. This address space is an abstraction of the physical memory. It implements virtual memory. It consists of logical memory addresses, which are numbered from address 0 upwards and it is independent from the storage technology used and the existing expansion options.

5. Explain, why virtual memory helps to better utilize the main memory.

Processes do not need to be located in one piece inside the main memory. Therefore, the external fragmentation of the main memory is not a problem.

6. Describe what mapping is.

The virtual memory is mapped to the physical memory.

7. Describe what swapping is.

The process of relocating data from the main memory to the SDD/HDD and back.

8. Name the component of the CPU that is used to implement virtual memory.

Memory Management Unit (MMU).

9. Describe the function of the component from subtask 8.

Virtual memory addresses are translated into physical memory addresses by the CPU using the MMU.

10. Name a virtual memory concept.

Paging.

11. Name the sort of fragmentation that does occur with the concept of subtask 10.

Internal fragmentation. It can only occur in the last page of each process.

12. Explain the purpose of the Page-Table Base Register (PTBS).

It stores the address where the page table of the current process starts.

Content: Topics of slide set 5 Page 3 of 11

13. Explain the purpose of the Page-Table Length Register (PTLR).

It stores the length of the page table of the current process.

14. Explain what a page fault exception causes to occur.

A process tries to access a page, which is not located in the physical main memory.

15. Describe the reaction of the operating system when a page fault exception occurs.

The operating system handles the page fault exception by executing these steps:

- Allocate the page by using the controller and the device driver on the swap memory (SSD/HDD).
- Copy the page into a free page of the main memory.
- Update the page table.
- Return control to the process. The process next tries to execute again the instruction that caused the page fault.
- 16. Explain what an access violation exception or general protection fault exception causes to occur.

A process tried to access a virtual memory address, which it is not allowed to access.

17. Describe the consequence (effect) of an access violation exception or general protection fault exception.

In some legacy Windows operating systems, segmentation faults often caused system crashes and resulted in a blue screen. In Linux, the signal SIGSEGV is returned as a result.

18. Describe the content of the kernelspace.

The operating system kernel and kernel extensions (drivers).

19. Describe the content of the userspace.

The currently running process, which is extended with swap memory (Windows: page file).

Exercise 4 (Memory Management)

Please mark for each one of the following statements, whether the statement is true or false.

Content: Topics of slide set 5 Page 4 of 11

1.	Real mode is s	suited for multitasking systems.
	☐ True	⊠ False
2.	-	mode, each process is executed in its own copy of the physical which is protected from other processes.
	⊠ True	☐ False
3.	When static p	artitioning is used, internal fragmentation occurs.
	⊠ True	☐ False
4.	When dynami	c partitioning is used, external fragmentation cannot occur.
	☐ True	⊠ False
5.	With paging,	all pages have the same length.
	⊠ True	\square False
6.	One advantage	e of long pages is little internal fragmentation.
	☐ True	⊠ False
7.	A drawback of	f short pages is that the page table gets bigger.
	⊠ True	☐ False
8.	When paging physical memory	is used, the MMU translates the logical memory addresses into bry addresses.
	⊠ True	\square False
9.	Modern opera paging.	ting systems (for x86) operate in protected mode and use only
	⊠ True	\square False

Exercise 5 (Page Replacement Strategies)

1. Why is it impossible to implement the optimal replacement strategy OPT?

Because it is not possible to predict the future and therefore the future request sequence is unknown.

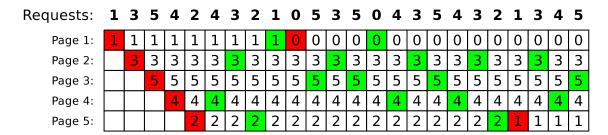
2. Perform the access sequence with the replacement strategies Optimal, LRU, LFU and FIFO once with a cache with a capacity of 4 pages and once with 5 pages. Also calculate the hit rate and the miss rate for all scenarios.

Content: Topics of slide set 5 Page 5 of 11

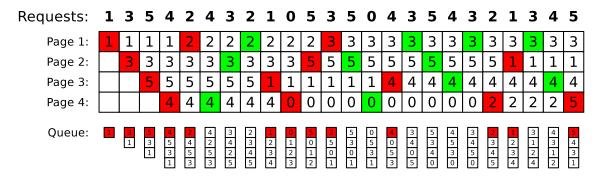
Optimal replacement strategy (OPT):

Requests:	1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3	4	5
Page 1:	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Page 2:		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Page 3:			5	5	2	2	2	2	2	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Page 4:				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

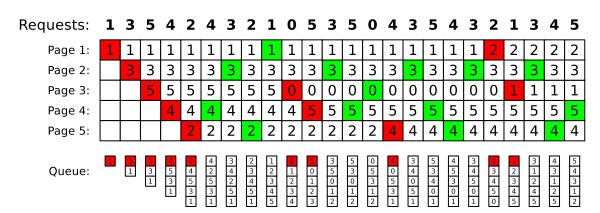
Hitrate: 15/24 = 0.625Missrate: 9/24 = 0.375



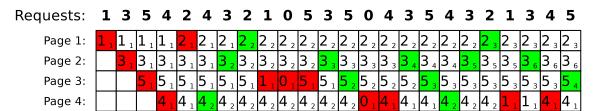
Hitrate: 17/24 = 0.7083333Missrate: 7/24 = 0.2916666 Replacement strategy Least Recently Used (LRU):



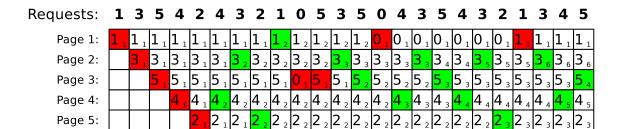
Hitrate: 11/24 = 0.4583333%Missrate: 13/24 = 0.5416666%



Hitrate: 14/24 = 0.583333%Missrate: 10/24 = 0.416666% Replacement strategy Least Frequently Used (LFU):

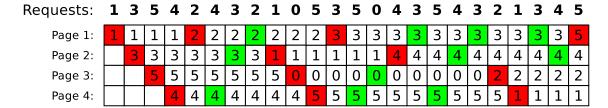


Hitrate: 12/24 = 0.5Missrate: 12/24 = 0.5



Hitrate: 15/24 = 0.625Missrate: 9/24 = 0.375

Replacement strategy FIFO:



Hitrate: 11/24 = 0.4583333Missrate: 13/24 = 0.5416666

Requests:	1	3	5	4	2	4	3	2	1	0	5	3	5	0	4	3	5	4	3	2	1	3	4	5
Page 1:	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Page 2:		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1	1
Page 3:			5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	3	3
Page 4:				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5
Page 5:					2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

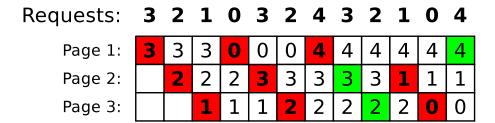
Hitrate: 15/24 = 0.625Missrate: 9/24 = 0.375

3. What is the key message of Laszlo Belady's anomaly?

FIFO produces worse results for certain access patterns with increased memory.

Content: Topics of slide set 5 Page 8 of 11

4. Show Belady's anomaly by performing the access sequence with the replacement strategy FIFO once with a cache with a capacity of 3 pages and once with 4 pages. Also calculate the hit rate and the miss rate for both scenarios.



Hitrate: 3/12 = 25%Missrate: 9/12 = 75%

Requests:	3	2	1	0	3	2	4	3	2	1	0	4
Page 1:	3	3	3	3	3	3	4	4	4	4	0	0
Page 2:		2	2	2	2	2	2	3	3	Э	3	4
Page 3:			1	1	1	1	1	1	2	2	2	2
Page 4:				0	0	0	0	0	0	1	1	1

Hitrate: 2/12 = 16.66%Missrate: 10/12 = 83.33%

Exercise 6 (Time-based Command Execution, Sorting, Environment Variables)

1. Create in your home directory a directory NotImportant and write a cron job, which erases the content of the directory NotImportant every Tuesday at 1:25 clock am.

The output of the command should be appended to a file EraseLog.txt in your home directory.

\$ mkdir ~/NotImportant

\$ crontab -e

Insert these lines:

25 1 * * 2 rm -rfv /home/USERNAME/NotImportant/* >>
/home/USERNAME/EraseLog.txt

2. Write a cron job, which appends a line at a file Datum.txt with the following format (but with the current values) every 3 minutes between 14:00 to 15:00 clock on every Tuesday in the month of November:

\$ crontab -e

Insert these lines:

3. Write an at-job, which outputs at 17:23 today a list of the running processes.

```
You may have to install the command line tool at first.

With Debian/Ubuntu this works with:

$ sudo apt update && sudo apt install at

With CentOS/Fedora/RedHat this works with:

$ sudo yum install at
```

\$ at 1725 today

Insert these lines:

ps -r

4. Write an at-job, which outputs at December 24th at 8:15 am the text "It's christmas!"

```
$ at 0815 DEZ 25
```

Insert these lines:

```
echo "It's christmas!"
```

5. Create in your home directory a file Kanzler.txt with the following content:

```
Willy
          Brandt
                     1969
Angela
          Merkel
                     2005
Gerhard
          Schröder
                     1998
KurtGeorg Kiesinger 1966
Helmut
          Kohl
                     1982
Konrad
          Adenauer
                     1949
Helmut
          Schmidt
                     1974
Ludwig
          Erhard
                     1963
```

```
$ echo "Willy Brandt 1969" >> ~/Kanzler.txt
$ echo "Angela Merkel 2005" >> ~/Kanzler.txt
```

```
$ echo "Gerhard
                  Schröder
                            1998" >> ~/Kanzler.txt
$ echo "KurtGeorg Kiesinger 1966" >> ~/Kanzler.txt
$ echo "Helmut
                  Kohl
                            1982" >> ~/Kanzler.txt
                            1949" >> ~/Kanzler.txt
$ echo "Konrad
                  Adenauer
$ echo "Helmut
                  Schmidt
                            1974" >> ~/Kanzler.txt
                            1963" >> ~/Kanzler.txt
$ echo "Ludwig
                  Erhard
```

6. Print out the file Kanzler.txt sorted by the first names.

```
$ sort ~/Kanzler.txt
```

7. Print out the file Kanzler.txt sorted by the third letter of the last names.

```
$ sort -k+2.4 ~/Kanzler.txt
```

8. Print out the file Kanzler.txt sorted by the year of the inauguration.

```
$ sort -k3 ~/Kanzler.txt
```

9. Print out the file Kanzler.txt backward reverse sorted by the year of the inauguration and redirect the output into a file Kanzlerdaten.txt.

```
$ sort -k3 -nr ~/Kanzler.txt > ~/Kanzlerdaten.txt
```

10. Create with the command export an environment variable VAR1 and assign it the value Testvariable.

```
$ export VAR01=Testvariable
```

11. Print out the value of VAR1 in the shell.

```
$ printenv VAR01
```

12. Erase the environment variable VAR1.

\$ unset VAR01