FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Security Research Group
Dept. of Computer Science

ITiCSE 2011

acm

# Combining Memory Management and Filesystems in an Operating Systems Course

Hans-Georg Eßer (h.g.esser@cs.fau.de)
Security Research Group, Dept. of Computer Science, Friedrich-Alexander-University Erlangen-Nuremberg

## Abstract

In a typical Operating Systems course, **memory management** and **filesystems** are treated as completely separate topics.
Since these two topics share many common principles and ideas, we have tried a combined approach where each concept has been explained only once and then shown in filesystem and memory management settings.

## Our Contribution

- Most works on the didactics of Operating Systems look at practical aspects, e. g., attempts to let students implement parts of an operating system.
- We look at the presentation of the core theory contents and how courses may profit from altering the standard order of presentation.
- The broader idea is to identify (abstract) concepts which re-occur in several areas of Computer Science. Instead of letting these concepts surface every now and then, we lay the focus on them and show their various applications simultaneously.

## Summary of Results

- Students performed better in the memory management and filesystems parts of the final exam (in comparison to the overall performance and compared with the previous year's course).
- Results from a questionnaire show that students liked the combined presentation and that it did not cause confusion.

## Course Contents

1. Introduction / Overview
   - Tasks of Filesystems
   - Tasks of Memory Management
2. Contiguous / Non-Contiguous Allocation
3. Internal and External Fragmentation
4. Contiguous Allocation
   - Dynamic Partitioning
   - Handling Free Space: Linked Lists, Bit Maps
   - Allocation: First-Fit, Best-Fit, Worst-Fit, Quick-Fit, Buddy System
   - MM: Code Relocation, Memory Protection
5. Non-Contiguous Allocation
   - Blocks (FS) and Pages (MM)
   - MM: Segmentation
   - FS: Indirection with Multi-Layer Index Blocks
   - FS: Unix Filesystems, Linux VFS, Ext3 FS
   - MM: Virtual Memory (Paging): Translation Look-Aside Buffer, Inverted Page Tables, Multi-Layer Paging, Page Faults, Page Replacement Strategies
   - Principle of Locality
6. FS: Swapping

## References

[1] Eßer, H.-G., *Treating Memory Management and Filesystems as One Topic*, University of Erlangen, Dept. of Computer Science, Technical Reports, CS-2011-04, April 2011. `http://www.opus.ub.uni-erlangen.de/opus/volltexte/2011/2507/pdf/tr_cs_2011_04.pdf`

[2] Eßer, H.-G., Website with Operating Systems courses' content: `http://hm.hgesser.de/`

[3] Eßer, H.-G., Introduction to Memory Management and Filesystems (draft), in: Betriebssysteme, Lecture Notes, Munich Univ. of Applied Sciences, 2009. `http://hm.hgesser.de/bs-ss2009/skript/skript-bs-kap06.pdf`

[4] Munich University of Applied Sciences: `http://hm.edu/`

[5] Hatfield, D. J., Experiments on Page Size, Program Access Patterns, and Virtual Memory Performance, IBM Journal of Research and Development 16, 1, 58-66 (1972).

## The Setting

- Course: **Introduction to Operating System Principles**, Munich University of Applied Sciences [4], Summer Semesters 2008, 2009
- Final exams taken by 27 students (2008) and 16 students (2009)
- 2009 course evaluation form filled out by 10 students

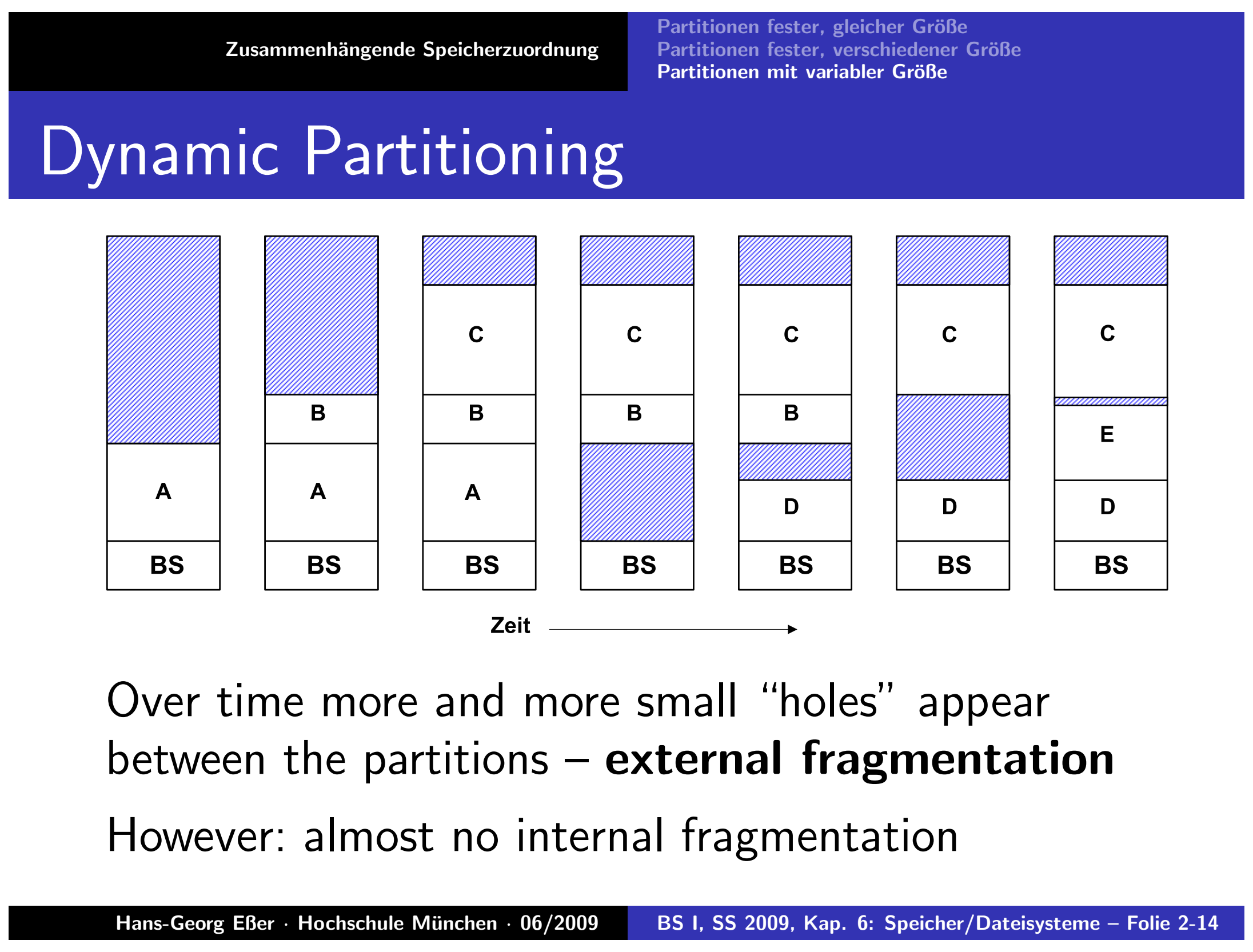
## Topics We Combined

- Simple allocation schemes, such as **fixed size partitioning**, where each process is given a fixed amount of memory or a file can use a fixed amount of disk space
- Management of **free block lists** (filesystem) and **free page frame lists** (memory management) via bit strings (free/used)
- **Internal and external fragmentation** which can occur both in filesystems and in simple memory management schemes
- **Indirection** (for keeping track of a list of blocks used by a file) vs. paging with **split page tables**.
- The **principle of locality** which applies to all areas of data access

## Examples

Two examples show what kinds of topics were taught in a combined way:

- With **Dynamic Partitioning** the operating system allocates and releases storage (either in RAM or on disk) dynamically. Since this simple scheme requires contiguous allocation, repeated (de-) allocations lead to external fragmentation, i. e., unusable tiny free areas (left picture).
- The **Principle of Locality** states that after accessing some memory location (which may be in a CPU cache, in RAM, or on disk) it is likely that further accesses in the immediate neighborhood will soon occur (right picture; source of image: Hatfield 1972 [5]).



Zusammenhängende Speicherzuordnung

Partitionen fester, gleicher Größe
Partitionen fester, verschiedener Größe
Partitionen mit variabler Größe

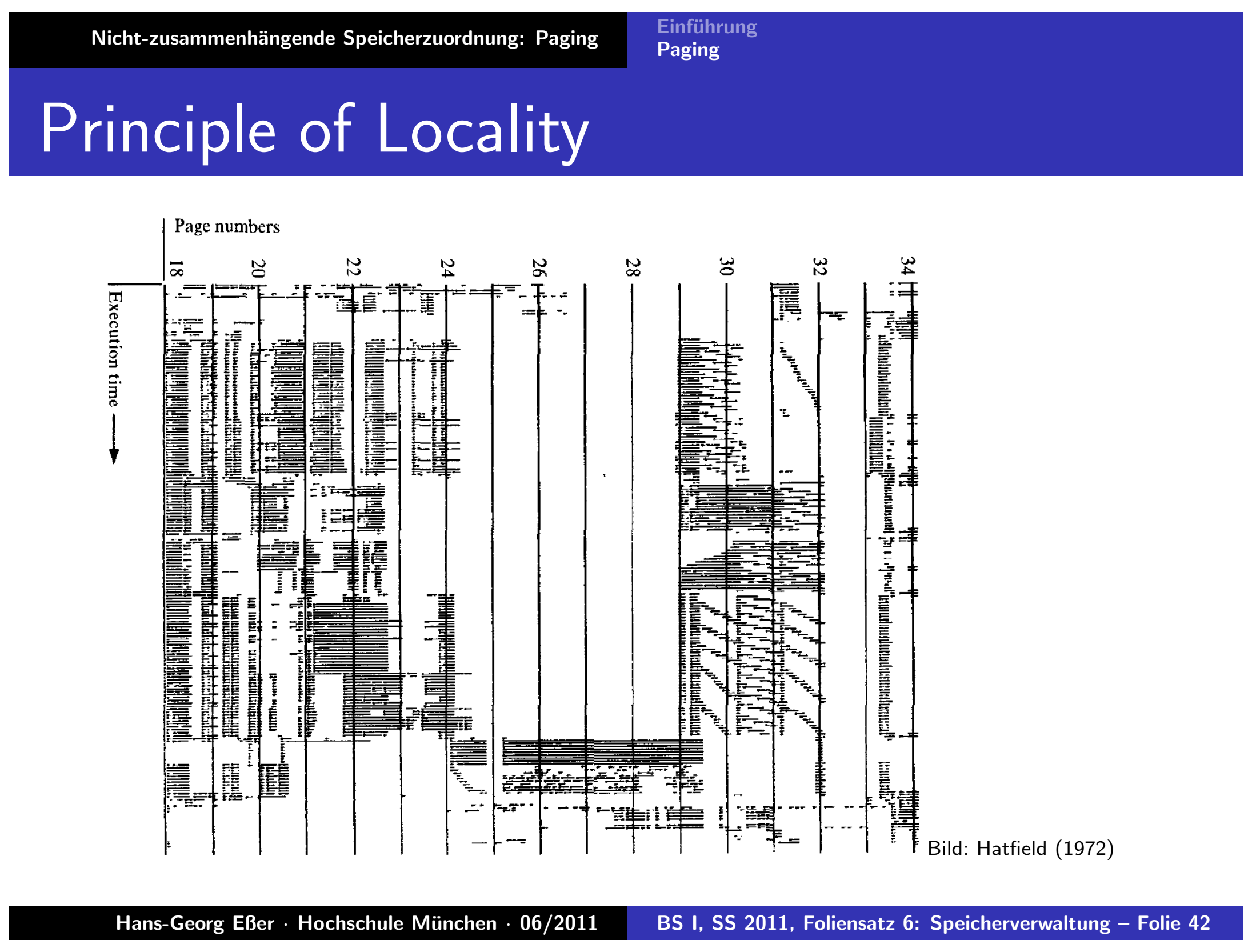
### Dynamic Partitioning

Over time more and more small "holes" appear between the partitions – **external fragmentation**

However: almost no internal fragmentation

Hans-Georg Eßer · Hochschule München · 06/2009

BS I, SS 2009, Kap. 6: Speicher/Dateisysteme – Folie 2-14



Nicht-zusammenhängende Speicherzuordnung: Paging

Einführung
Paging

### Principle of Locality

Bild: Hatfield (1972)

Hans-Georg Eßer · Hochschule München · 06/2011

BS I, SS 2011, Foliensatz 6: Speicherverwaltung – Folie 42

These slides were taken from our 2009 course's slide set on filesystems and memory management (translated).

## Evaluation I: Final Exam Results

Comparison of Students' Exam Results:

| | | Old | New | | Old | New | Change |
|---|---|---|---|---|---|---|---|
| Memory Management (MM) | $P_M$ | 48.20% | 52.59% | $P_M/P_T$ | 80.56% | 82.43% | + 2.32% |
| Filesystems (FS) | $P_F$ | 58.80% | 72.92% | $P_F/P_T$ | 98.28% | 114.29% | +16.30% |
| FS+MM | $P_{FM}$ | 52.59% | 62.39% | $P_{FM}/P_T$ | 87.90% | 97.79% | +11.25% |
| overall | $P_T$ | 59.83% | 63.80% | $P_T/P_T$ | 100.00% | 100.00% | — |

$P_M$, $P_F$: Average percentage of points gained in MM / FS related questions, $P_T$: Average percentage of points gained in overall exam. Percentages in right part of table were derived from those in left part by dividing each value by the same column's overall value.

## Evaluation II: Student Questionnaire

Special questions for the modified course with combined MM/FS treatment:

| Question / Statement | Avg. | 1 | 2 | 3 | 4 | 5 | n/a |
|---|---|---|---|---|---|---|---|
| 1. Combined treatment of the topics makes sense | 1.6 | 50% | 40% | 10% | 0% | 0% | 0% |
| 2. Frequent change between properties of filesystems and memory management is confusing | 4.1 | 0% | 10% | 10% | 40% | 40% | 0% |
| (*Negation of the above statement*) | *1.9* | *40%* | *40%* | *10%* | *10%* | *0%* | *0%* |
| 3. The combination made it easy to understand that many concepts from one topic translate to the other topic | 1.4 | 60% | 40% | 0% | 0% | 0% | 0% |
| 4. More topics should be combined this way | 2.8 | 20% | 10% | 40% | 10% | 10% | 10% |

Scale: 1 = agree completely; 5 = disagree completely.

## Suggestions for Further Research

- Repeat the experiments with a larger group of students
- Test combined teaching of other related topics, e. g., atomicity in database transactions vs. atomicity of synchronization operations (for mutexes, semaphores)