Betriebssystemtechnik

Adressräume: Trennung, Zugriff, Schutz

POSIX Speicherverwaltung

SS 2026

Wolfgang Schröder-Preikschat / Volkmar Sieh

Lehrstuhl für Informatik 4
Systemsoftware

Friedrich-Alexander-Universität
Technische Fakultät

Gliederung

Strukturen

```
1   struct inode {
2     ...
3     struct mm_reg *first;
4     struct mm_reg *last;
5     ...
6   };
```

```
1   struct file {
2       ...
3       struct inode *inode;
4       ...
5   };
```

```
1   struct thread_files {
2     ...
3     struct file *file[NFILES];
4     ...
5   };
```

```
1   struct mm_reg {
2     struct mm_reg *prev;
3     struct mm_reg *next;
4
5     struct mm_reg *proc_prev;
6     struct mm_reg *proc_next;
7
8     struct inode *inode;
9     off_t offset;
10    size_t size;
11    char *vaddr;
12    ...
13  };
14
15  struct thread_mm {
16    struct hw_mmu_ctxt mmu_ctxt;
17    ...
18    struct mm_reg *reg_first;
19    struct mm_reg *reg_last;
20    ...
21  };
```

```
1   struct thread {
2     struct thread *prev;
3     struct thread *next;
4     ...
5     struct thread_mm *mm;
6     struct thread_file *files;
7     ...
8   };
```

Strukturen / Variablen

```
1   struct page {
2     struct page *free_prev;
3     struct page *free_next;
4     struct page *hash_prev;
5     struct page *hash_next;
6
7     int refcnt;
8     enum { CLEAN, DIRTY, ... } state;
9
10    struct inode *inode;
11    off_t offset;
12
13    ...
14  };
```

```
1   struct thread *first_thread;
2   struct thread *last_thread;
3
4   struct page page[NPAGES];
5   struct page *free_first;
6   struct page *free_last;
7   struct page *hash_first[1024];
8   struct page *hash_last[1024];
```

+ Page-Tabelle ;-)

Funktionen

- File-I/O-Funktionen:

```
1   int open(const char *filename, int flags);
2   int close(int fd);
3
4   ssize_t read(int fd, void *buf, size_t len);
5   ssize_t write(int fd, const void *buf, size_t len);
6
7   int ftruncate(int fd, off_t len);
```

- Memory-Management-Funktionen:

```
1   void *mmap(void *addr, size_t len, int prot, int flags, int fd, off_t offset);
2   int munmap(void *addr, size_t len);
3
4   int brk(void *addr);
5   void *sbrk(size_t inc);
6
7   pid_t fork();
8   void exit(int retval);
9
10  void mm_fault(void *addr, int prot); /* Page-Fault-Handler */
```

Gliederung

Schluss

Danke für die Mitarbeit! Danke für die gute Atmosphäre!

Viel Erfolg für das weitere Studium!