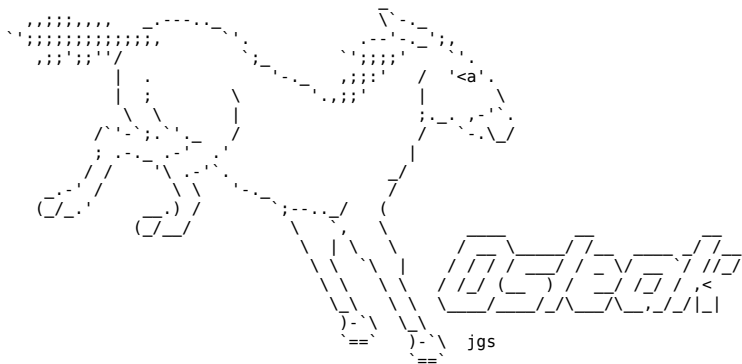


# OSy: Osleak

Karel Fišer, Vojtěch Horký, Karel Jakubec



# Kernel threads

- `thread_t` is integer
- red-black tree (taken from Linux)
- round-robin scheduler without priorities
- Solaris-like processes

# Memory management

Same data structure in

- frame allocator
- virtual memory allocator

The structure

- two RB-trees sharing same data nodes
- unused nodes in pools

Features

- logarithmic search by address or size
- effective memory usage

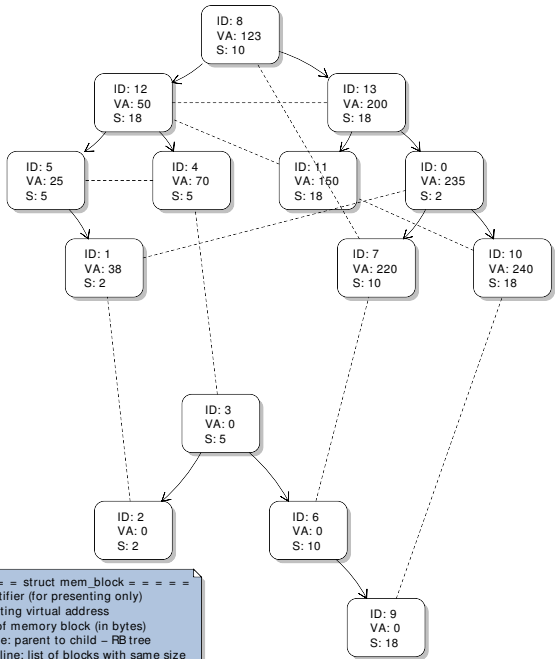
## The memory block struct

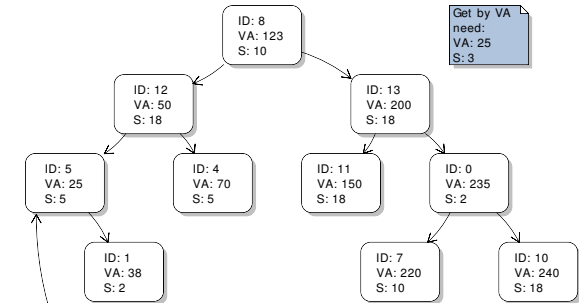
```
struct mem_block {
    /** Node of RB tree (by size or address). */
    struct rb_node node_;

    /** List of blocks of equal size. */
    struct list_head list_;

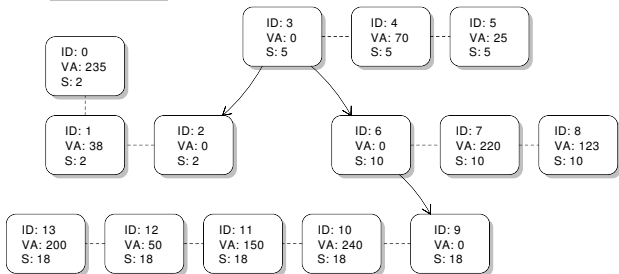
    /** Size of block. */
    size_t size_;

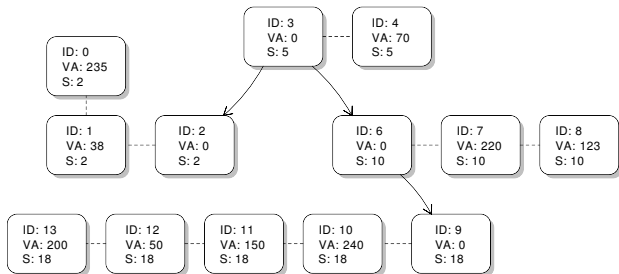
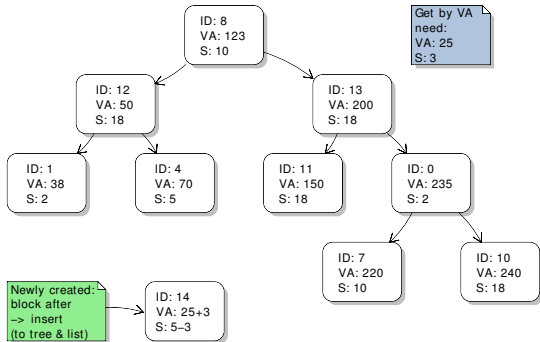
    /** Starting address of block. */
    void * addr_;
};
```

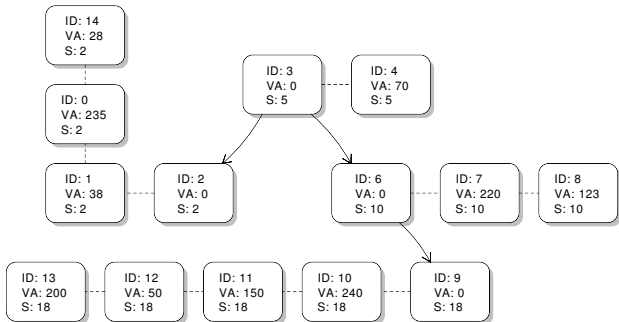
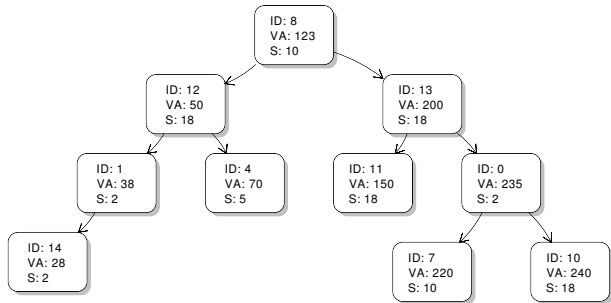




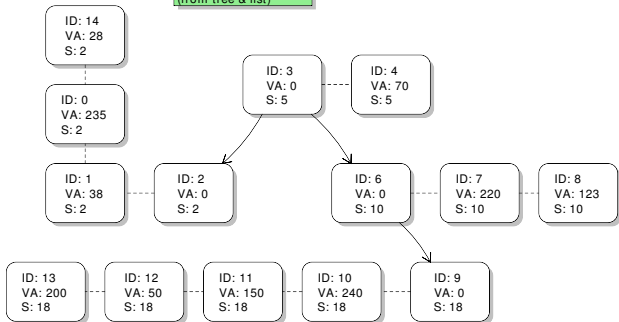
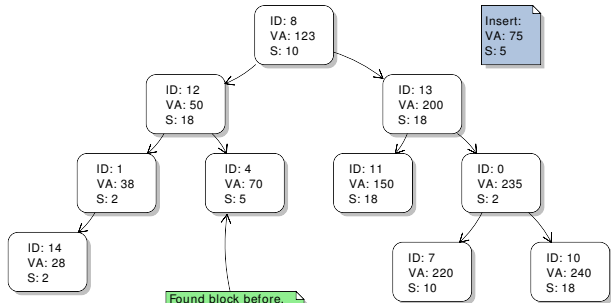
Found ID 5  
VA & S - OK  
-> delete ID 5  
(from tree & list)

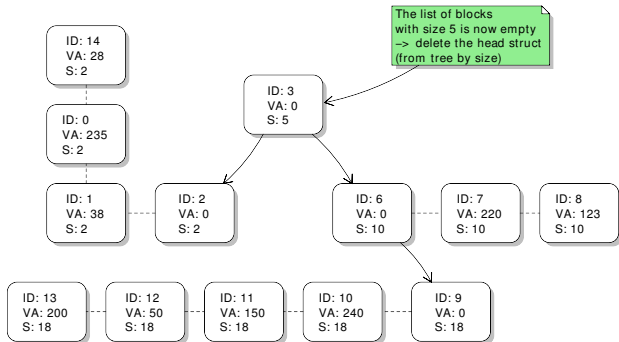
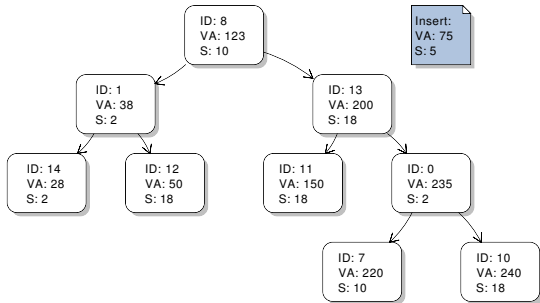


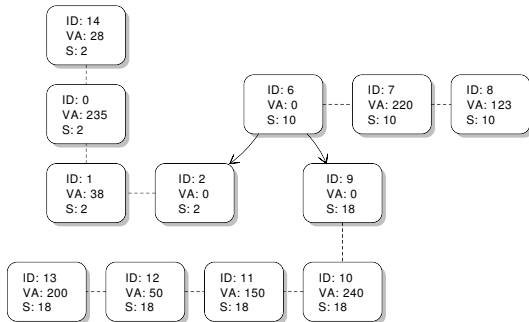
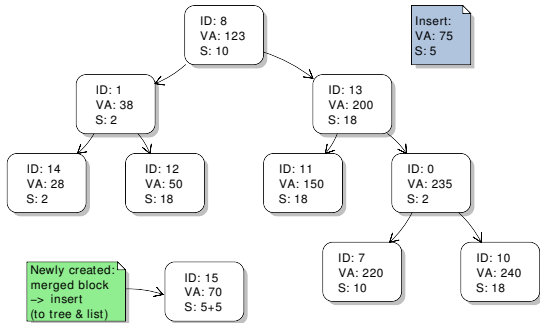


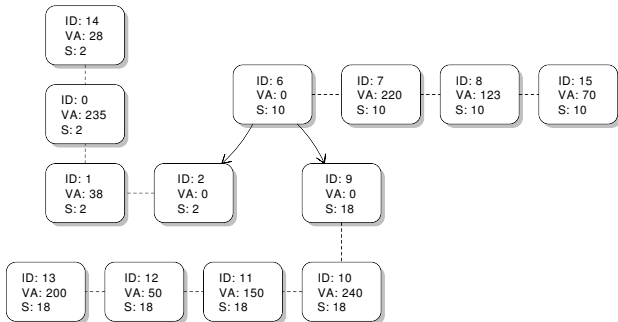
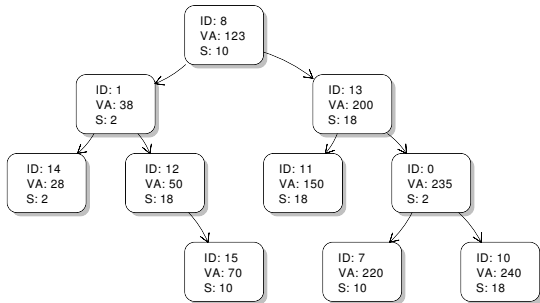


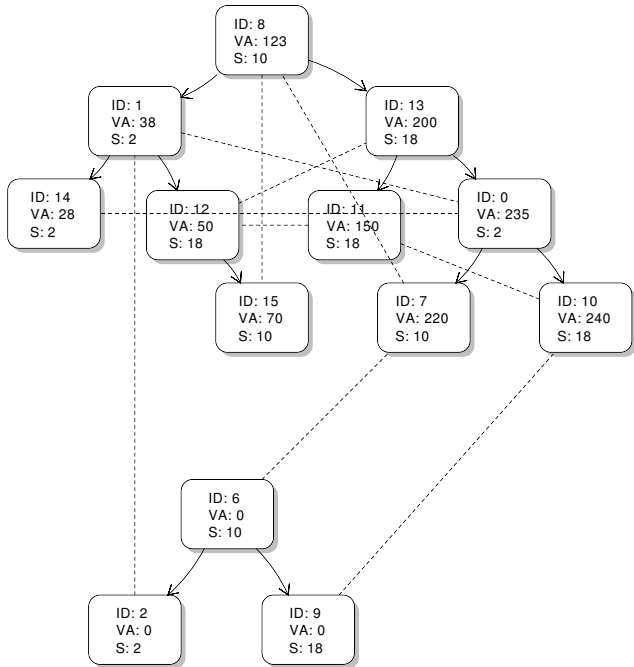












# Virtual memory management

- two-level paging
- ASIDs
- only 2 reserved TLB entries

## The problem: creating a new syscall

- symbolic name & number
- kernel handler
  - actual handler (parameter mangling)
  - `switch`
- userspace caller
  - library function (wrapper)
  - actual syscall (parameter mangling)

## The solution: single .def file

```
DECLARE_SYSCALL(PRINTSTRING,  
                IN_PARAM(char *, s, 1)) {  
  
    VERIFY_USERSPACE_POINTER(s);  
  
    puts(s);  
  
    return EOK;  
}
```



## The solution (cont.)

`.def file`  $\xrightarrow{\$(CC) -E + sed(1)}$

- symbolic names `#defines`
- kernel handlers
- kernel `switch` with type casting
- userspace wrapper to `syscall()`

# Piquancies

- standard header files
- pthread API in userspace
- `brk()` & `sbrk()` system calls
  - Doug-Lea for userspace malloc
- unit testing script