

# Operating systems

## Workshops 2

Today we're going to use [Valgrind](#). It's extremely helpful tool during development and testing phase of software production. Students are encouraged to read more about it.

### *Exercise 1*

Use following command: `sysctl -a |& grep ^vm | sort`, to list all Linux kernel parameters associated with virtual memory. They're documented [here](#). What's the purpose of following parameters: `oom_kill_allocating_task` and `swappiness`?

### *Exercise 2*

Use following command: `grep ^Vm /proc/$$/status`, to print current process address space information. Conduct similar experiment on other processes that you own (find PID with `ps -f -u $USER`). Look at `VmPeak` and `VmSize` fields, give their meaning. What are the sizes of data, code, stack segments? How much memory does OS use to store page tables that describe address space of the process?

### *Exercise 3*

Use `free` command to show memory statistics. What is the meaning of `shared buffers` and `cached` columns? Can you find factual size of used physical memory?

### *Exercise 4*

Let's have another look at user space (`malloc / free`) memory management. We already know, that such an allocator need to somehow obtain memory block from an OS, in order to satisfy program requests. Write a very simple application in C language, that will allocate a few blocks of memory. Show with `strace` tool, which system call is used to increase memory size available for the allocator.

*Exercise 5*

Open following document: [Massif: a heap profiler](#). Compile a program from chapter 9.2.1 listing, and carry out instructions from points 9.2.2 to 9.2.6. How Massif can help in development process?

*Exercise 6*

Run a web browser (i.e. Firefox). Using Massif analyze its memory usage over the time. Perform several actions that are supposed to allocate and release memory (e.g. tab open / close). Do they immediately relinquish memory to operating system?