

# OS08b: Virtual Memory with Linux






([Usage hints](#) for this presentation)

Computer Structures and Operating Systems 2023  
Dr. Jens Lechtenbörger ([License Information](#))





## Speaker notes

- To toggle these notes, press `v`
  - If a slide contains audio, notes might show transcript
- Press `?` for key bindings (in particular, `a`, `o`, `n`, `p`, `Ctrl-Shift-f`)
- Presentations support two different PDF formats, see [usage notes](#) 
  - Both hyperlinked on index page
    - Concise PDF format (replace `.html` and whatever follows in [address bar](#)  with `.pdf`)
    - Print browser view to PDF (add `?print-pdf` after `.html`, then print to PDF; [suggested settings](#) )
- If you find the amount of outgoing links to be distracting, see [usage notes](#) 
  - Add `?hideLinks` (maybe with a number) after `.html`
- See [usage notes](#)  for other non-obvious features



# 1. Looking at Memory with Linux

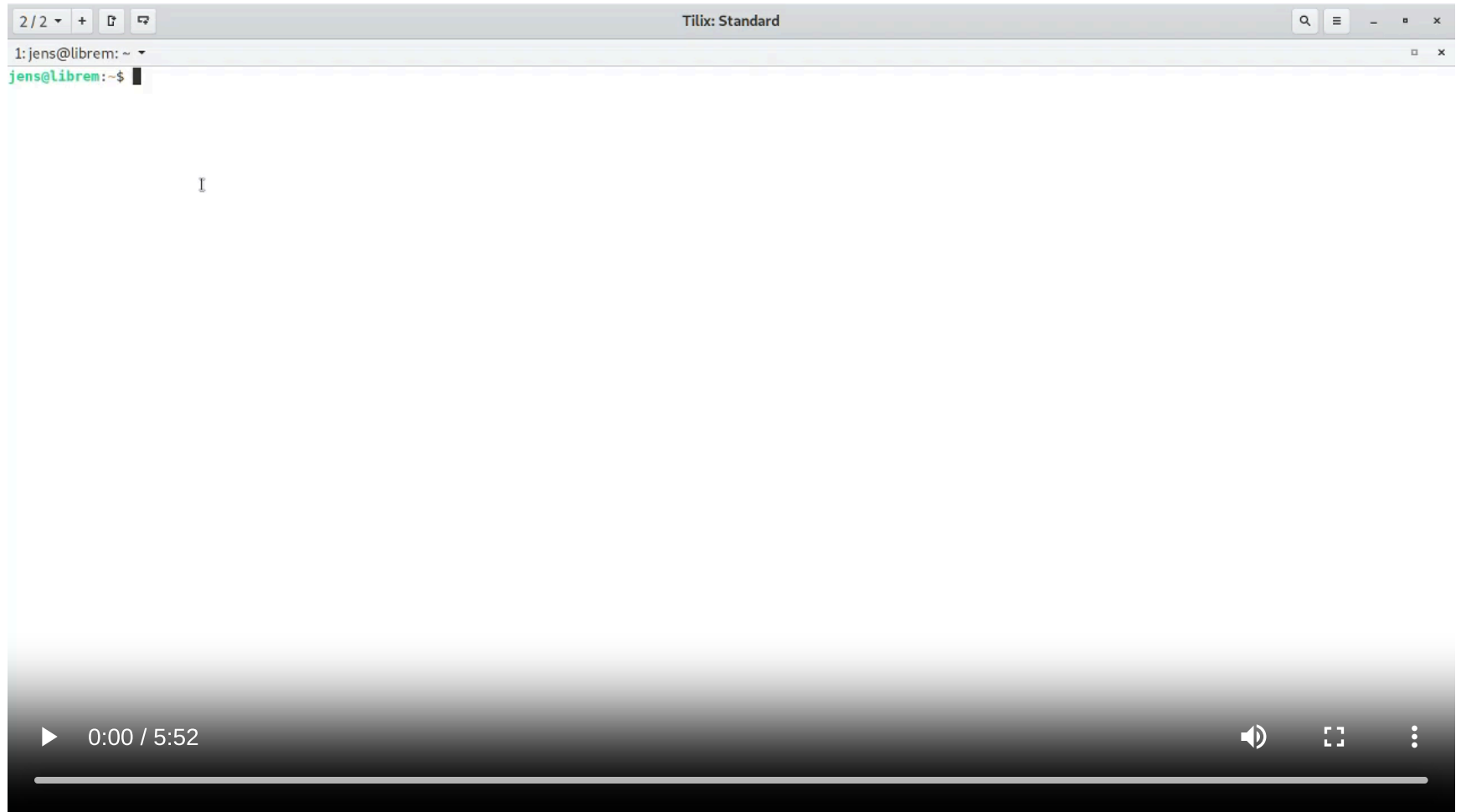
(Specifics of Linux are not part of learning objectives; however, the following illustrates [shared memory](#), and `/proc` will be revisited in other presentations.)



# 1.1. Linux Kernel: `/proc/<pid>/`

- `/proc` is a pseudo-filesystem
  - See <https://man7.org/linux/man-pages/man5/proc.5.html>
    - (Specific to Linux kernel; incomplete or missing elsewhere)
  - “Pseudo”: Look and feel of any other filesystem
    - Sub-directories and files
    - However, files are no “real” files but meta-data
  - Interface to internal **kernel data structures**
    - One sub-directory per process ID
    - OS identifies process by integer number
    - Here and elsewhere, `<pid>` is meant as **placeholder** for such a number

# 1.1.1. Video about /proc





This video, “Looking at /proc” by [Jens Lechtenbörger](#), shares the presentation's license terms, namely [CC BY-SA 4.0](#).

The video shows some aspects of the `/proc` filesystem related to memory management, which are described in more abstract form on subsequent slides.

# 1.1.2. Drawing about /proc

an amazing directory: **/proc** JULIA EVANS @b0rk

<p>Every process on Linux has a PID (process ID) like 42.</p> <p>In <code>/proc/42</code>, there's a lot of VERY USEFUL information about process 42</p>	<p><b>/proc/PID/cmdline</b></p> <p>command line arguments the process was started with</p>	<p><b>/proc/PID/exe</b></p> <p>symlink to the process's binary <b>magic</b>: works even if the binary has been deleted!</p>
<p><b>/proc/PID/fd</b></p> <p>Directory with every file the process has open!</p> <p>Run <code>\$ls -l /proc/42/fd</code> to see the list of files for process 42.</p> <p>These symlinks are also magic &amp; you can use them to recover deleted files ♥</p>	<p><b>/proc/PID/stack</b></p> <p>The kernel's current stack for the process. Useful if it's stuck in a system call</p>	<p>and <b>more</b></p> <p>Look at</p> <p><b>man proc</b></p> <p>for more information!</p>
	<p><b>/proc/PID/maps</b></p> <p>List of process's memory maps. Shared libraries, heap, anonymous maps, etc.</p>	

/proc

Figure © 2018 Julia Evans, all rights reserved; from [julia's drawings](#).  
Displayed here with personal permission.

# 1.1.3. Drawing about **man** pages

man pages = awesome  
(sometimes. Quality may vary :))

JULIA EVANS  
@b0rk



I found out I can get documentation for programs (like grep) with **man grep**!



but that's not all!! lots of other things have man pages too!

man pages are split up into 8 sections  
① ② ③ ④ ⑤ ⑥ ⑦ ⑧

/usr/share/man/man5  
has section 5 on my machine.

GREAT →

- ① programs  
\$man grep  
\$man ls
- ③ C functions  
\$man 3 printf  
\$man fopen
- ⑤ file formats  
\$man sudoers  
for /etc/sudoers  
\$man proc
- ⑦ miscellaneous  
\$man 7 pipe  
\$man 7 symlink  
(these are cool!)
- ② system calls  
\$man sendfile
- ④ devices  
\$man null  
for /dev/null docs
- ⑥ games  
(not very useful)  
man sl is good if you have sl though
- ⑧ sysadmin programs  
\$man apt  
\$man chroot


Man pages are amazing

Figure © 2016 Julia Evans, all rights reserved; from [julia's drawings](#).  
Displayed here with personal permission.

## 1.2. Linux Kernel Memory Interface

- Memory allocation (and much more) visible under `/proc/<pid>`
- E.g.:
  - `/proc/<pid>/pagemap`: One 64-bit value per virtual page
    - Mapping to RAM or swap area
  - `/proc/<pid>/maps`: Mapped memory regions
  - `/proc/<pid>/smaps`: Memory usage for mapped regions
- Notice: Memory regions include **shared** libraries that are used by lots of processes

## 1.3. GNU/Linux Reporting: smem

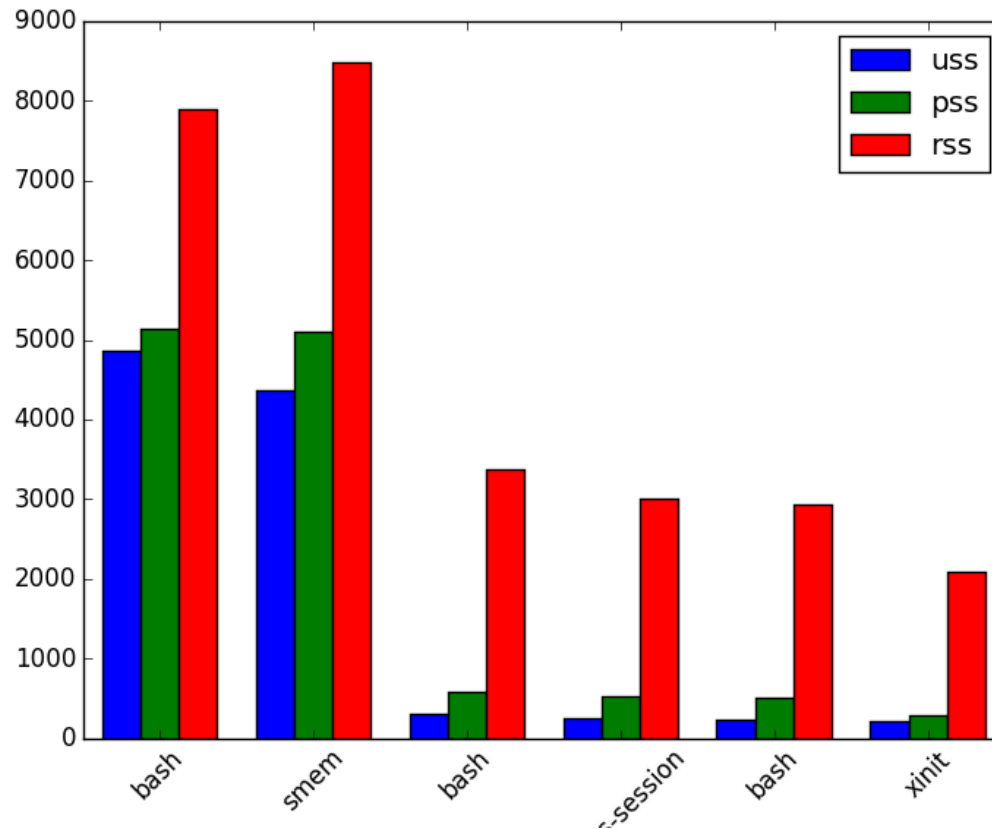
- User space tool to read smaps files: smem
  - See <https://linoxide.com/memory-usage-reporting-smem/> 
- Terminology
  - **Virtual set size** (VSS): Size of virtual address space
  - **Resident set size** (RSS): Allocated main memory
    - Standard notion, yet overestimates memory usage as lots of memory is shared between processes
      - Shared memory is added to the RSS of every sharing process
  - **Unique set size** (USS): memory allocated exclusively to process
    - That much would be returned upon process' termination
  - **Proportional set size** (PSS): USS plus “fair share” of shared pages
    - If page shared by 5 processes, each gets a fifth of a page added to its PSS

## 1.3.1. Sample smem Output

```
$ smem -c "pid command uss pss rss vss" -P "bash|xinit|emacs"
```

PID	Command	USS	PSS	RSS	VSS
765	/usr/bin/xinit /etc/X11/Xse	220	285	2084	15952
1390	/bin/bash -c libreoffice5.3	240	510	2936	13188
826	/bin/bash /usr/bin/qubes-se	256	524	3008	13204
750	-su -c /usr/bin/xinit /etc/	316	587	3368	21636
1251	bash	4864	5136	7900	26024
2288	/usr/bin/python /usr/bin/sm	5272	6035	9432	24688
1145	emacs	90876	93224	106568	662768

## 1.3.2. Sample smem Graph



```
smem --bar pid -c "uss pss rss" -P "bash|xinit"
```

"Screenshot of smem" under CC0 1.0; from GitLab

# License Information

This document is part of an [Open Educational Resource \(OER\)](#) course on Operating Systems. [Source code and source files are available on GitLab](#) under [free licenses](#).

Except where otherwise noted, the work “OS08b: Virtual Memory with Linux”, © 2017-2022 [Jens Lechtenbörger](#), is published under the [Creative Commons license CC BY-SA 4.0](#).

*No warranties are given. The license may not give you all of the permissions necessary for your intended use.*

In particular, trademark rights are *not* licensed under this license. Thus, rights concerning third party logos (e.g., on the title slide) and other (trade-) marks (e.g., “Creative Commons” itself) remain with their respective holders.



