WWU MÜNSTER

ERCIS

OS08b: Virtual Memory with Linux

(Usage hints for this presentation)

Computer Structures and Operating Systems 2023

Dr. Jens Lechtenbörger (License Information)

Data Science: Machine Learning and Data Engineering (Prof. Gieseke)
Dept. of Information Systems

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

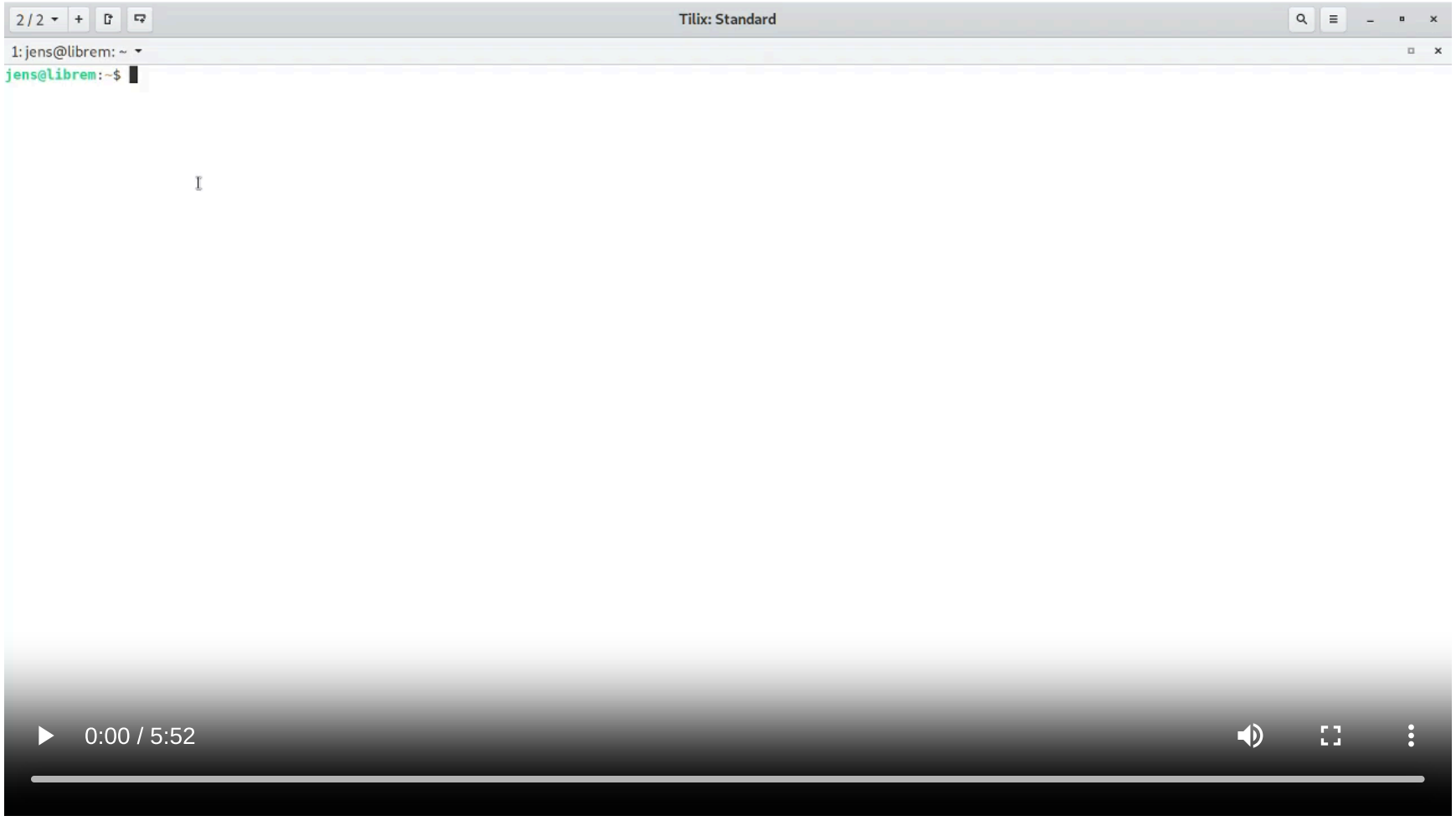
1. Looking at Memory with Linux

(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`

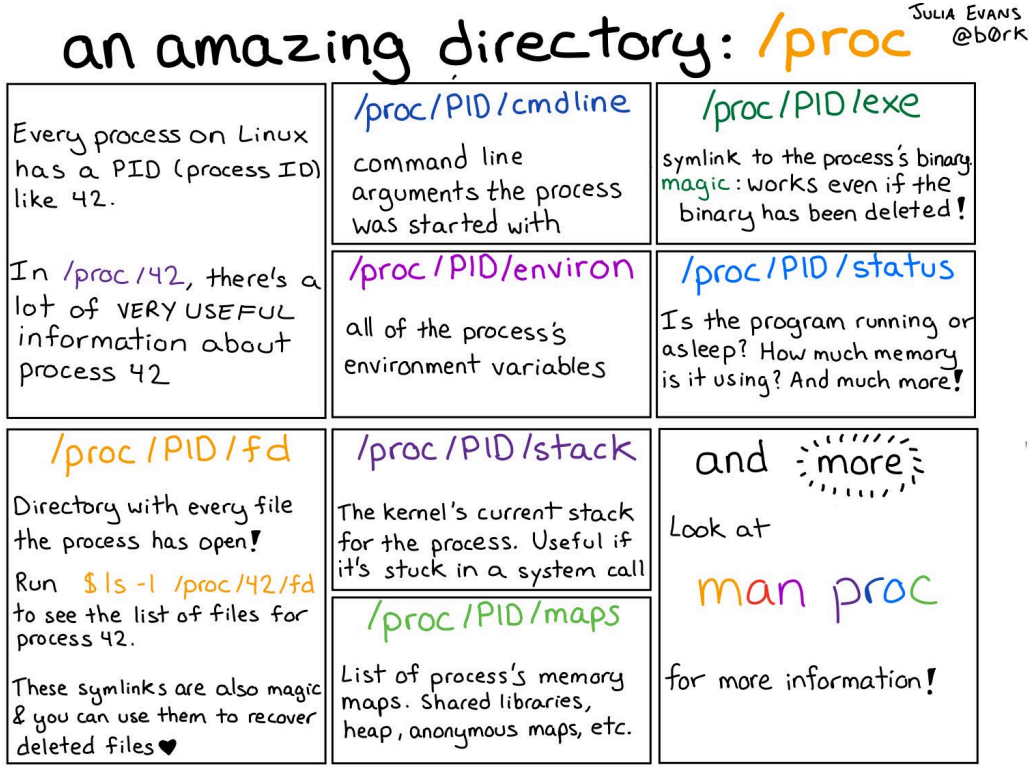


Speaker notes

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



/proc

1.1.3. Drawing about man pages

Man pages are amazing

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

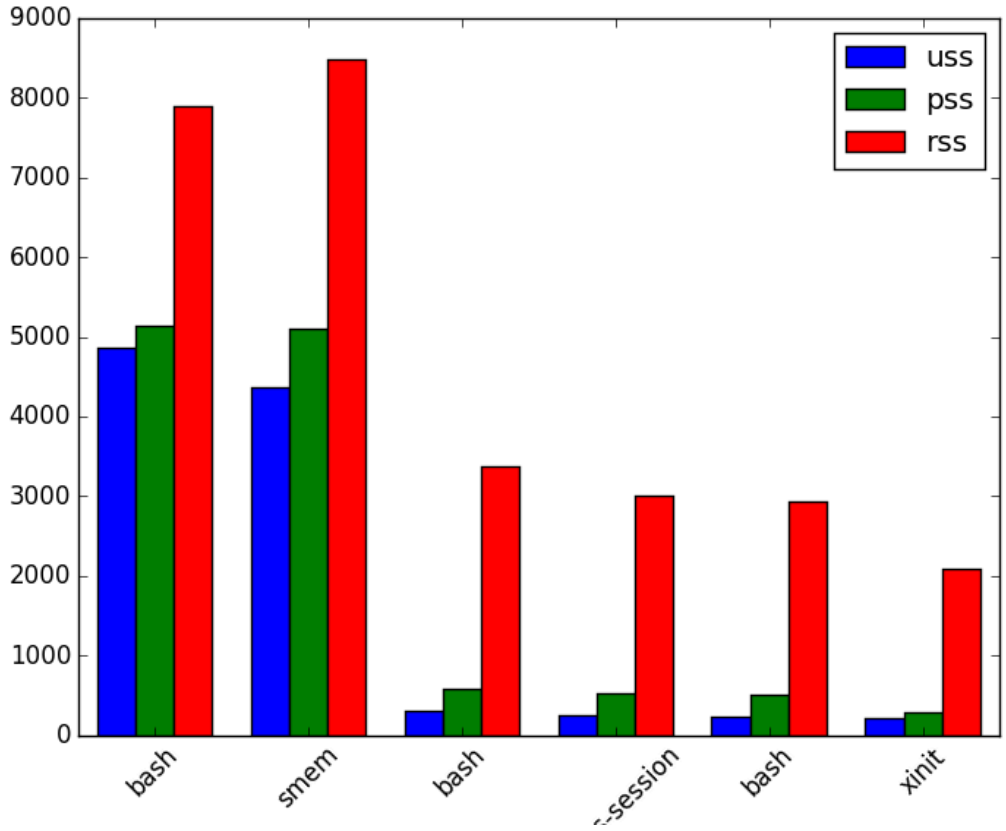
1.3. GNU/Linux Reporting: `smem`

- 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"
```

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.

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