Implementing application-level ballooning

E0253 programming assignment

Memory overcommitment

- Virtual vs. physical memory
  - Physical memory is often overcommitted

Memory overcommitment

- Virtual vs. physical memory
  - Physical memory is often overcommitted

Image credit: https://www.reddit.com/r/memes/comments/he7c52/hulk_smash/

Overcommitment techniques

- ❑ Swapping
- ❑ Ballooning
- ❑ Memory compression
- ❑ Remote memory (disaggregated systems)

Swapping and page replacement

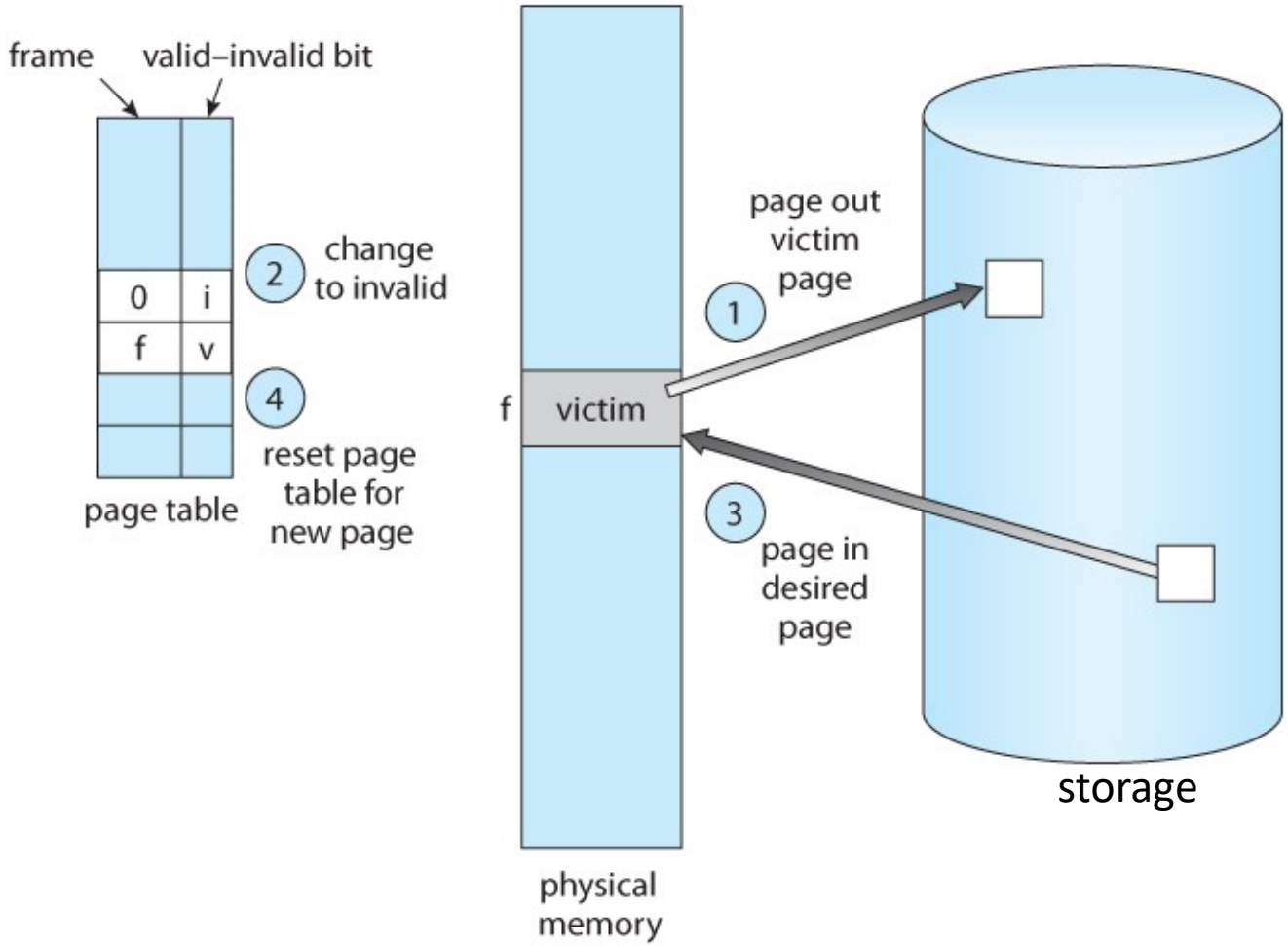


Image credit: https://image.slidesharecdn.com/osppt4unit-120825101537-phpapp02/95/os-swapping-paging-segmentation-and-virtual-memory-43-728.jpg?cb=1345890022

Memory ballooning

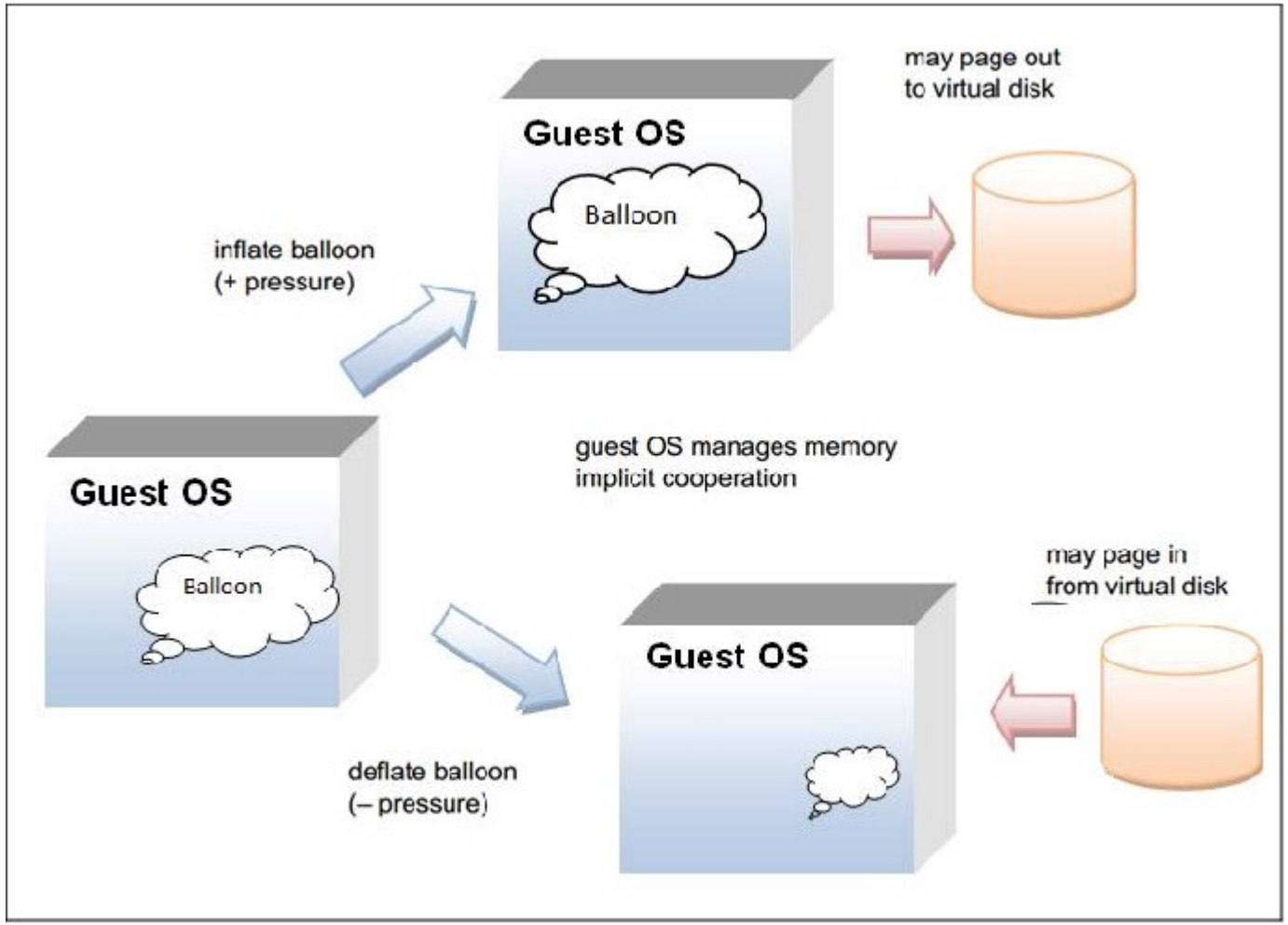
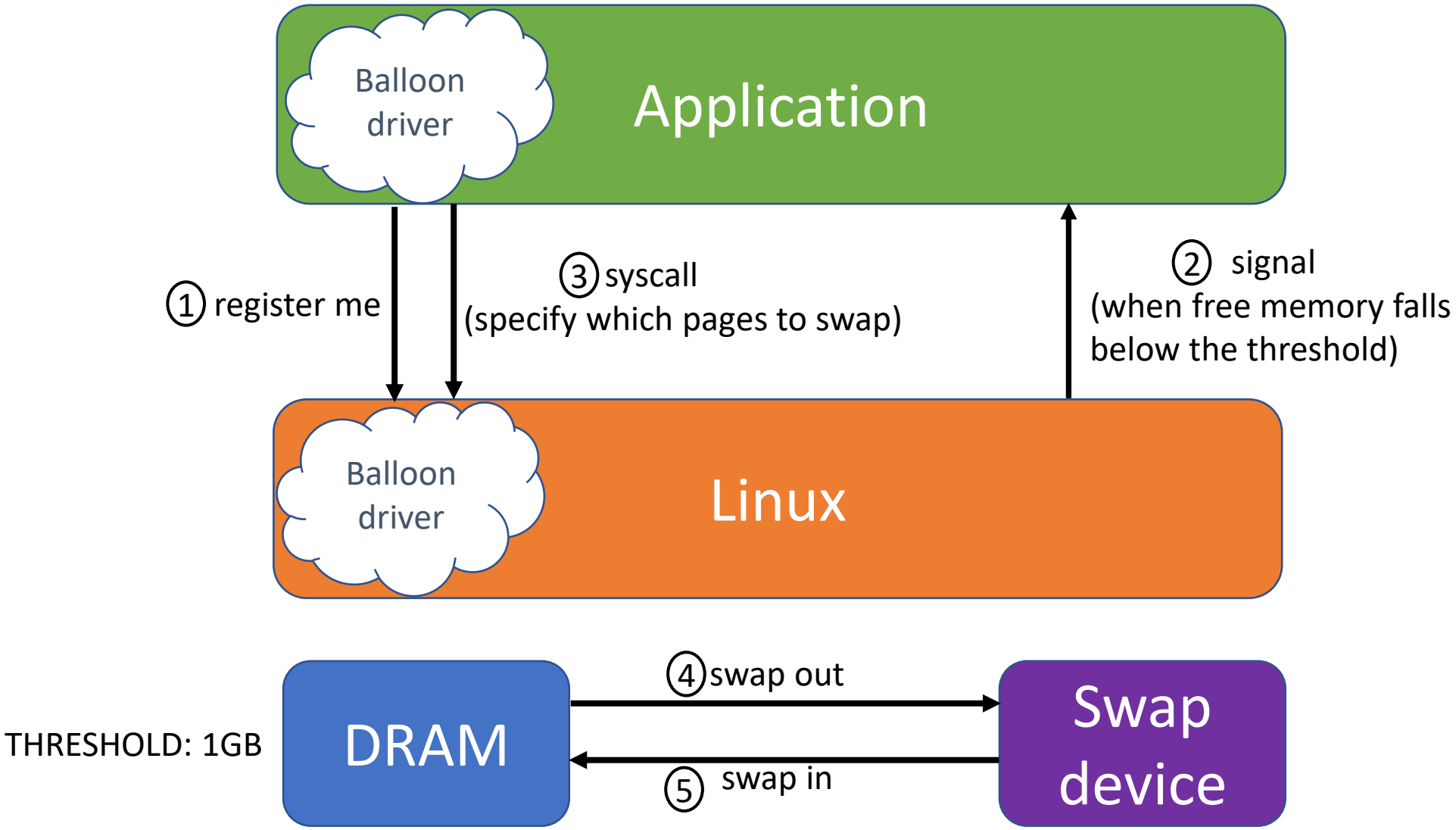


Image credit: https://d3i71xaburhd42.cloudfront.net/24c3f115efcdb4d52cb0111554fcd18a068313bf/2-Figure2.1-1.png

Assignment: Application-level ballooning



Part-1: Basic setup

Kernel space	User space
• Develop a balloon driver • Ensure that only registered applications' pages can be swapped (disable swapping for others after an application has registered itself) • Identify when free memory in the system is below a pre-configured threshold (use 1GB as the threshold) • Signal the registered application when free memory is below the threshold (add a new signal SIGBALLOON)	• Register application with the kernel's ballooning driver • Implement a handler for the SIGBALLOON signal. Track how many times SIGBALLOON has been received.

Part-2: Policies and mechanisms

Kernel space (mechanism)	User space (policy)
• Implement mechanism(s) to swap out pages specified by the application. • Pages should be swapped out only if free memory is below the threshold.	• Implement a page replacement policy to identify pages that are good swapping candidates. • Request the kernel to swap out the selected pages. • You will need to examine the access patterns of the application to devise a suitable policy. However, we will not provide you the source code. Instead, you will use existing kernel interfaces to study the page access patterns. • Try to minimize swap-related IO activities. Assume temporal and spatial locality.

Part-3: To be announced

This will likely involve multi-processing

Getting started

Linux kernel (version 5.11.5)

https://www.kernel.org/

User space application

https://github.com/csl-iisc/e0253-os-2021

More details

https://github.com/csl-iisc/e0253-os-2021/blob/main/assignment.pdf