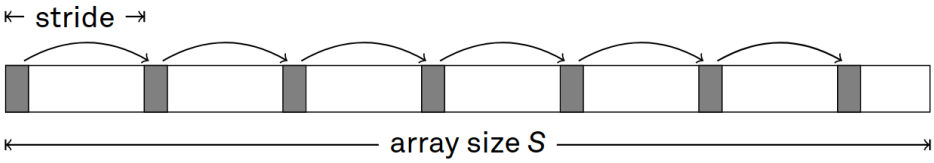
Data Processing on Modern Hardware
Assignment 1

Handout: 26^{st} April 2023
Due date: 03^{th} May 2023

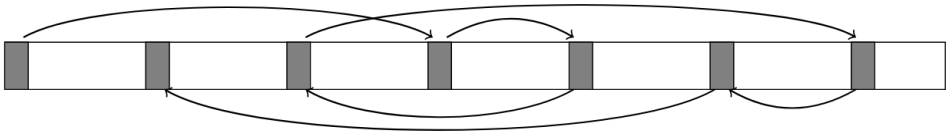
Set-up

The main goal of this assignment is to measure the latency of each level of the memory hierarchy of your machine. To achieve that, a micro-benchmark is needed that accesses the system's memory with specific access patterns. The elements of the array are accessed in one of the following patterns:

1. Sequential



2. Reverse sequential
    - Like the sequential access pattern, but starts with the last element
    - Can also use a stride to add distance between the accessed elements
3. Random (from the perspective of the hardware prefetcher)



4. Blocked
    - The array contains pointers to other arrays
    - The arrays with the actual data are not guaranteed to be next to each other in memory

Tasks

a Use the provided assignment template and implement the missing methods in cache_access.c that is located in src.

b Run the benchmark for the three access patterns with different values for d (the stride) and S (the size of the array) on your system (e.g., $d \in [8B, \ldots, 16kB]$ and $S \in [4kB, \ldots, 512MB]$) and describe/plot the measured values.

c Based on the measurements answer the following questions.

- What observations do you make?
- Does the experiment match your expectations / hypothesis?
- How can you explain the observed behavior?
- What conclusions can be drawn from your hardware based on the results?

Note that if you have access to different computers (with different architectures (e.g., AMD vs. Intel vs. ARM), or older vs. newer hardware, or different cache and memory configurations. etc.), you are welcome to experiment with them and report the obtained numbers.

Submission guidelines

This homework has a duration of one week. Fork the repository and commit your changes in the git. Your submission should include a short report (report.pdf) with some visualizations that answers the questions.

Hint

In the article "What Every Programmer Should Know About Memor" by Ulrich Drepper, you can find a lot of information for implementing the micro-benchmark and interpreting the results.

Have a look at the built-in functions of GCC or inline assembly to add prefetching.