Project Title: Memory Latency Profiling and Software Prefetching on the Raspberry Pi 5

Project Goal: To explore the memory hierarchy of the Raspberry Pi 5 and evaluate the impact of software prefetching on program performance. The project will profile memory latency across cache levels, implement a simple software prefetching technique, and analyze its effect on performance metrics.

Hardware Requirements: - Raspberry Pi 5 (ARM Cortex-A76 CPU, LPDDR5 memory) - Cooling solution (heatsink/fan) to prevent thermal throttling - Optional: USB power meter for energy impact measurement

---

Software Requirements: - Operating System: Raspberry Pi OS - Programming Language: C or C++ (for benchmarks and prefetching code) - Tools: - `perf` (performance counters, cache/memory profiling) - Python + matplotlib (for plotting results) - Optional: `valgrind` or `cachegrind` for memory analysis

---

Project Deliverables:

1. **Memory Latency Profiler**
2. Microbenchmark code to measure:
    - Cache latency (L1, L2)
    - DRAM latency
    - Bandwidth vs. working set size
3. Produce plots of latency vs. working set size (“memory mountain”)

4. Deliverable: Code + graph showing Pi 5 memory hierarchy

5. **Software Prefetching Implementation**

6. Choose a benchmark kernel (matrix multiplication, BFS, etc.)
7. Write two versions:
    1. Baseline (no prefetching)
    2. Prefetch-optimized (manual software prefetching)
8. Measure:
    - Execution time
    - Instructions per cycle (IPC)
    - Cache misses

9. Deliverable: Source code + table/graphs comparing performance

10. **Analysis and Report**

11. Compare performance of baseline vs. prefetching
12. Discuss:
    - When prefetching helps or hurts

    - Effect of memory stride and working set size
    - Insights about Pi 5 memory hierarchy
13. Deliverable: Written report (including plots and observations)

Optional Extension: - Implement an adaptive prefetcher: - Detect access stride at runtime - Adjust prefetch distance dynamically - Compare against static prefetching - Deliverable: Extended analysis showing dynamic prefetcher benefits (if implemented)

Summary: - Characterize memory latency and cache hierarchy on Pi 5 - Implement and evaluate software prefetching - Analyze performance improvements and limitations - Produce graphs, tables, and a report demonstrating findings