Lab 3 - Cache and I/O Programming over Raspberry Pi

This assignment was locked Oct 7 at 11:59pm.
6% of your final grade
Individual lab, not a team lab, no collaboration is allowed!

Lab 3 consists of two sections. Both use C/C++ as the programming language. In section 1, you will use the performance analyzing tool perf to measure the cache miss rate of your program. In section 2, you will practice how to directly manipulate GPIO registers from Raspberry Pi OS. Please refer to the lab slides and video for more details.

Section 1 (50%): Monitor Cache Misses of Your Code

Please write C/C++ code to multiple two N by N matrices. Please follow the (element data type, matrix size) specified in the table below to

1. Test your program with randomly generated matrices and analyze the cache misses using perf. Record the runtime (s) and L1 data cache miss rate (%) of your program for each test. Fill the table with your measurements.
2. Is the cache miss rate of your implementation < 0.1%? If not, can you meet this constraint by optimizing your code? (Hint: try swapping the order of your loops)

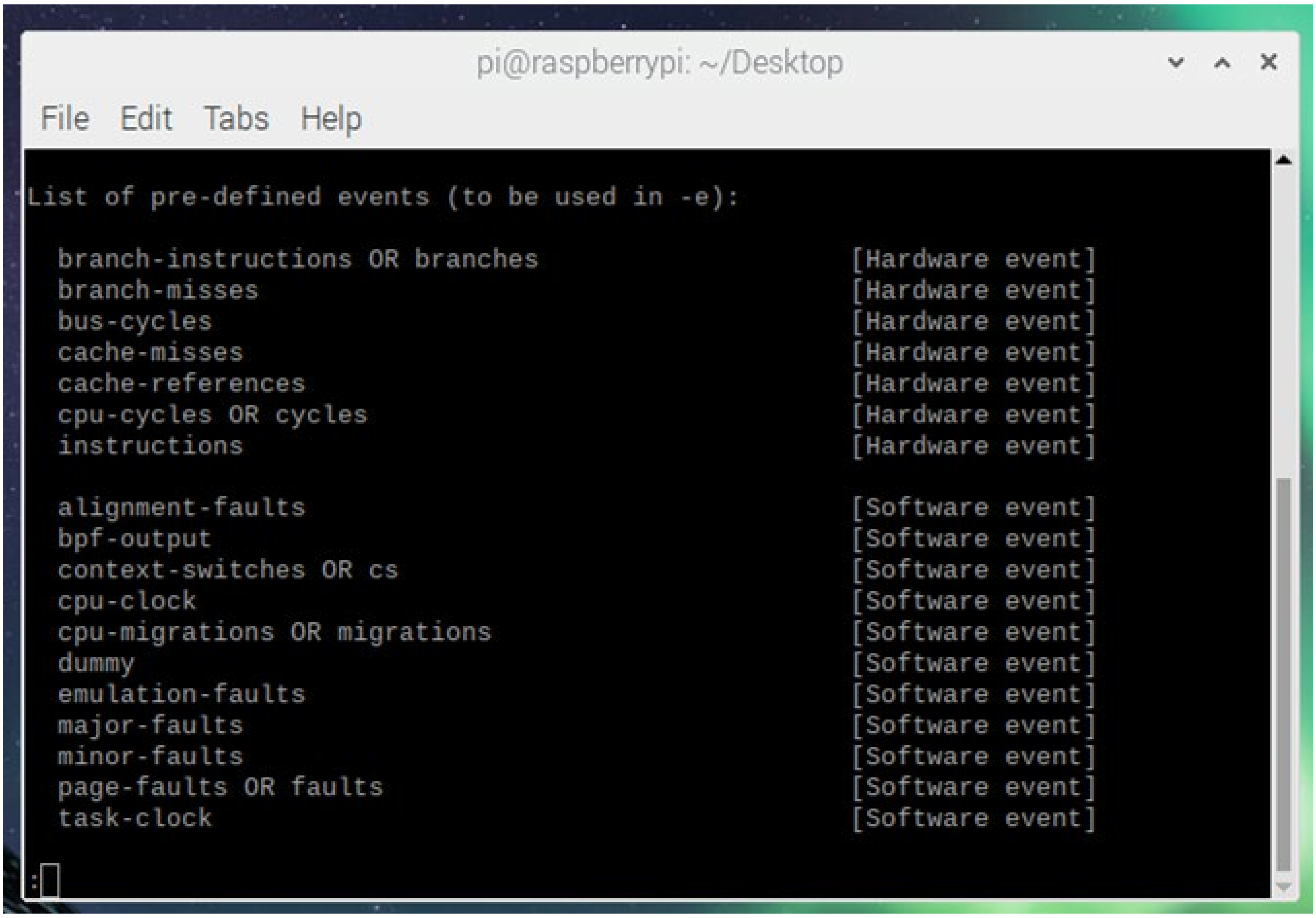
Configuration	int, N=256	int, N=512	int, N=1024
Runtime (s)			
Cache miss rate (%)			

You can use:
1. malloc() to allocate memory blocks.
2. rand() to generate random numbers and then cast to your desired data type.
3. clock() for runtime measurement. You may need to take average from multiple tries.

Jeremy Love
ECE 1175
Lab 3
10/7/2022

Section 1:

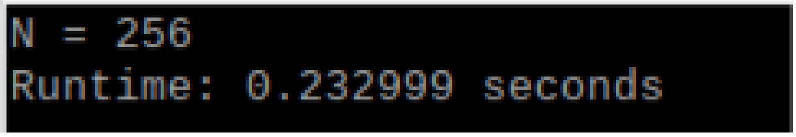
Note: My board did not have the required perf commands to measure cache misses. I optimized my program via a timer written into my code.



```
List of pre-defined events (to be used in -e):

  branch-instructions OR branches                    [Hardware event]
  branch-misses                                      [Hardware event]
  bus-cycles                                         [Hardware event]
  cache-misses                                       [Hardware event]
  cache-references                                   [Hardware event]
  cpu-cycles OR cycles                               [Hardware event]
  instructions                                       [Hardware event]

  alignment-faults                                   [Software event]
  bpf-output                                         [Software event]
  context-switches OR cs                             [Software event]
  cpu-clock                                          [Software event]
  cpu-migrations OR migrations                       [Software event]
  dummy                                              [Software event]
  emulation-faults                                   [Software event]
  major-faults                                       [Software event]
  minor-faults                                       [Software event]
  page-faults OR faults                              [Software event]
  task-clock                                         [Software event]

:
```

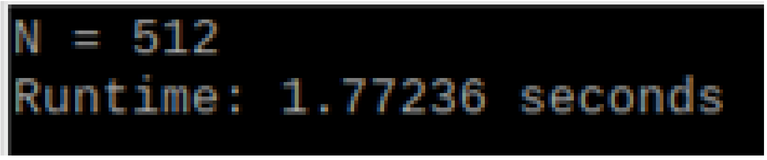
Results:

N = 256:
Runtime = 0.232999 seconds



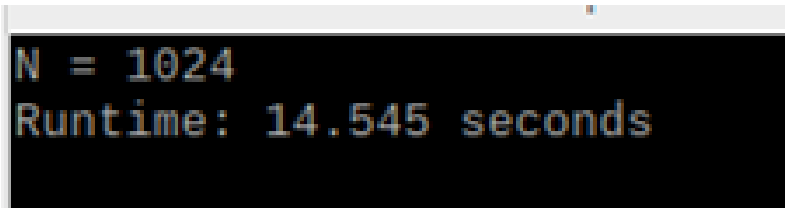
```
N = 256
Runtime: 0.232999 seconds
```

N = 512
Runtime = 1.77236 seconds



```
N = 512
Runtime: 1.77236 seconds
```

N = 1024
Runtime = 14.545 seconds



```
N = 1024
Runtime: 14.545 seconds
```