Week #5 Self-assessments: Timing & Profiling

Due Sep 24 at 11:59pm **Points** 15 **Questions** 15

Available until Sep 25 at 11:59pm **Time Limit** None **Allowed Attempts** 3

Instructions

This quiz is to help you review and self-assess your learning in this course. In addition, the objective is to motivate you to review concepts in a timely and diligent manner. These quizzes do count towards your final grade (see **Syllabus**). **Consequently, please do take these quizzes seriously**.

Attempt History

	Attempt	Time	Score
KEPT	Attempt 3	2 minutes	15 out of 15
LATEST	Attempt 3	2 minutes	15 out of 15
	Attempt 2	4 minutes	13 out of 15
	Attempt 1	53 minutes	12 out of 15

ⓘ Correct answers will be available on Sep 26 at 12am.

Score for this attempt: **15** out of 15
Submitted Sep 24 at 9:56pm
This attempt took 2 minutes.

Question 1

1 / 1 pts

Assume the timing (and other runtime characteristics) of a program called `./chatty` has to be measured. However, the program generates a lot of output which is hindering timing analysis. A good solution to measure the timing of the program would be:

- `/usr/bin/time ./chatty | /dev/null`

- `/usr/bin/time ./chatty < /dev/null`
- (selected) `/usr/bin/time ./chatty > /dev/null`
- `/usr/bin/time ./chatty`

Question 2

1 / 1 pts

A program was run on a very lightly-loaded machine with 8 cores. Given the following output from `/usr/bin/time`, what was the total time was spent running instructions as part of system calls?

```
4.17user 2.74system 0:05.10elapsed 374%CPU (0avgtext+0avgdata 22032maxresident)k
117184inputs+0outputs (0major+6143minor)pagefaults 0swaps
```

- 4.17 seconds
- (selected) 2.74 seconds
- 3.74 seconds
- 3.10 seconds

Question 3

1 / 1 pts

Assume the timing (and other runtime characteristics) of a program called `./magic` has to be measured, with command-line arguments `5 "test" ~/`. The correct bash shell command is:

- `time ./magic 5 "test" ~/`

- time 5 "test" ~/ ./magic
- /usr/bin/time ./magic 5 "test" ~/
- /usr/bin/time 5 "test" ~/ ./magic

Question 4

1 / 1 pts

A program was run on a very lightly-loaded machine with 8 cores. Given the following output from `/usr/bin/time`, how many threads were run on an average?

```
4.17user 2.74system 0:03.10elapsed 374%CPU (0avgtext+0avgdata 22032maxreside
nt)k
117184inputs+0outputs (0major+6143minor)pagefaults 0swaps
```

- 8
- 4
- 2
- 6

Question 5

1 / 1 pts

A program was run on a very lightly-loaded machine with 8 cores. Given the following output from `/usr/bin/time`, how long (as perceived by the user) did the program take to run?

```
4.17user 2.74system 0:03.10elapsed 374%CPU (0avgtext+0avgdata 22032maxreside
nt)k
117184inputs+0outputs (0major+6143minor)pagefaults 0swaps
```

- 2.74 seconds
- 0.31 seconds
- (selected) 3.10 seconds
- 4.17 seconds

Question 6

1 / 1 pts

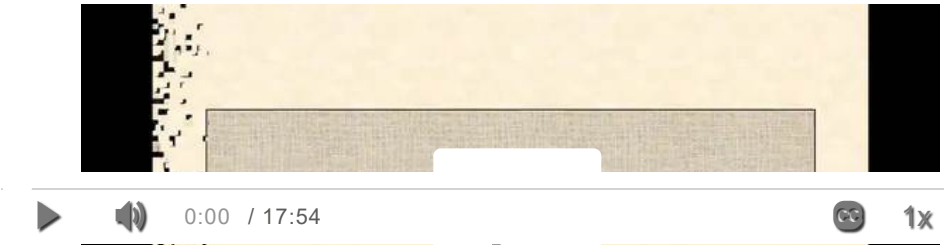
A program was run on a very lightly-loaded machine with 8 cores. Given the following output from `/usr/bin/time`, what was the total time was spent running instructions in userspace?

```
4.17user 2.74system 0:03.10elapsed 374%CPU (0avgtext+0avgdata 22032maxresident)k
117184inputs+0outputs (0major+6143minor)pagefaults 0swaps
```

- 3.10 seconds
- 2.74 seconds
- 374 seconds
- (selected) 4.17 seconds

The following questions are based on the content covered in the following presentation:

Profiling



Question 7

1 / 1 pts

The most accurate profilers that can be used to even analyze hardware-level behavior is

- (●) Hardware profiler
- ( ) Hardware-assisted profiler
- ( ) Hypervisor
- ( ) Simulators

Question 8

1 / 1 pts

Linux perf is a

- Software profiler
- (•) Hardware-assisted profiler
- Hypervisor-based profiler
- Hardware profiler

Question 9

1 / 1 pts

A company has developed a software profiler to measure the time taken to run a method call by simply adding a `Timer::start()` and `Timer::stop()` method around a method as shown below:

```
Timer::start();
methodCallToBeTimed();
Timer::stop();
```

The above approach is an example of

- Runtime instrumentation
- (•) Manual instrumentation
- Runtime injection
- Compiler assisted instrumentation

Question 10

1 / 1 pts

With Profile Guided Optimization (PGO) the gcc compiler instruments the generated binary to collect profile data. This approach is an example of

- [x] Compiler assisted instrumentation
- [ ] Runtime injection
- [ ] Binary translation
- [ ] Runtime instrumentation

Question 11

1 / 1 pts

A debugger can trace the operations of a program and even provide information about values of variables during runtime. This is accomplished by inserting extra instructions at breakpoints set by the user just before the program is executed for debugging. This is an example of

- [ ] Runtime injection
- [x] Runtime instrumentation
- [ ] Compiler assisted instrumentation
- [ ] Binary translation

Question 12

1 / 1 pts

The Java Virtual Machine (JVM) uses the Just-In-Time (JIT) compilation approach to dynamically generate instructions for the CPU from a Java program. This enables the JVM to profile a program at runtime by introducing additional instructions during JIT compilation. This approach is an example of

- [ ] Binary translation

- (selected) Runtime injection
- Compiler assisted instrumentation
- Runtime instrumentation

Question 13

1 / 1 pts

A key disadvantage of a software profiler is that

- They cannot be used for short methods
- They cannot be used on the OSC cluster
- They cannot be used for special types of instructions
- (selected) Instrumentation may skew behaviors of programs

Question 14

1 / 1 pts

Linux perf collects profiles of a program by

- Uses a container to trace program operations
- Instrumenting the program before it is run
- (selected) Sampling CPU counters at a given frequency
- Instrumenting the program while is is being run

Question 15

1 / 1 pts

A key disadvantage of Linux perf is that

- It cannot provide details for recursive methods
- It cannot be used to profile I/O
- It cannot trace pointer operations
- (selected) It cannot profile rare events

Quiz Score: **15** out of 15