

**Question 3: Know your system: Find a multicore environment where you can use at least two cores.**

**Investigate the environment and answer the following:**

1. What is the model, make and core count of your CPU?
  - a. Model : Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz
  - b. Make : Intel
  - c. Core count : 4
2. What are the L1, L2, L3, (L4?) cache, and memory sizes?
  - a. L1, L2 and L3 caches are different memory pools similar to the RAM in a computer. They were built in to decrease the latency to access data by the processor. The architecture they are built with also differs considerably. For e.g. the L1 cache is built using larger transistors and wider metal tracks, trading off space and power for speed. The higher-level caches are more tightly packed and use smaller transistors.
  - b. Memory : 8GB
  - c. L1i Cache : 32K
  - d. L1d Cache : 32K
  - e. L2 Cache : 256K
  - f. L3 Cache : 6144K
3. Clock speed, the MIPS and the MFLOPS rating for your CPU?
  - a. Clock speed: 2.5 GHz, can be overclocked to 3.5 GHz
  - b. MIPS : 4988
  - c. MFLOPS: Couldn't be found
4. Disk seek, latency, and transfer times (your local hard disk)?
  - a. Disk seek : 3s
  - b. Latency : 5.5 ms (5400 RPM)
  - c. Transfer rate : 6 Gb/s
5. How to know if your processor is 64b or 32b? How do you know if the processor supports "hyperthreading"?
  - a. We can find out by typing "`uname -a`" in the terminal.
  - b. ``lscpu`` specifies the number of threads per core. If this value is >1, the system is hyperthreaded. A system is hyperthreaded if it has more number of threads available than the number of cores.
6. OS version? Is 32 or 64 bit?
  - a. OS version : 16.04 ubuntu
  - b. 64 bit
7. Available VM size for user processes. Can it be changed?
  - a. The available VM size is practically infinite.
  - b. Yes, it can be changed. Only the total memory can be changed, not the memory allotted to each process.

8. Limits on the stack space, heap space and static area. Can they be changed?

- a. Stack space : 8192 KB
- b. Heap space : Couldn't find the exact size
- c. Static area : Depends on the bus size and virtual memory size

9. What is a memory leak and how can you detect one?

- a. A memory leak is a type of resource leak that occurs when a computer program incorrectly manages memory allocations in such a way that memory which is no longer needed is not released. A memory leak has symptoms similar to a number of other problems and generally can only be diagnosed by a programmer with access to the program's source code. Memory leak occurs when programmers create a memory in heap and forget to delete it. Memory leaks are particularly serious issues for programs like daemons and servers which by definition never terminate.
- b. Memory leaks can be detected using the Valgrind tool suite (<http://valgrind.org/docs/manual/quick-start.html>). Memory leaks can be detected using the Memcheck tool and the usage is mentioned in the link.

10. How can one cause a signal on stack overflow?

```
void sighandler(int signum) {
    printf("Process %d got signal %d\n", getpid(), signum);
    signal(signum, SIG_DFL);
    kill(getpid(), signum);
}

int main() {
    signal(SIGSEGV, sighandler);
    printf("Process %d waits for someone to send it\n", getpid());
    sleep(1000);
    return 0;
}
```

11. How can one tell how many page faults a process had?

- a. ``/usr/bin/time <executable>`` gives the number of page faults

12. How can one tell how much user, system and elapsed time a process used?

- a. By using the command ``time <executable>``

13. How can one time a procedure in a program? How accurate is this?

- a. We can use any one of the following functions:
  - i. `clock()`: This function gives the current CPU cycle. By calling this function at begin and end points, calculate the number of CPU cycles and divide by the number of cycles per second. This gives the time in seconds.
  - ii. `gettimeofday()`
  - iii. `clock_gettime()`
- b. The time obtained is accurate and the resolution is mention below:
  - i. `clock()`: If we follow the procedure mentioned above, we get the time in seconds.
  - ii. `gettimeofday()`: microsecond resolution

iii. `clock_gettime()`: nanosecond resolution