1. What is the model, make and core count of your CPU?
Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz (dmidecode)
Core Count: 2 (*Physical*) (dmidecode and lscpu)
CPU(s): 4   (*Logical*) (lscpu)

2. What are the L1, L2, L3, (L4?) cache, and memory sizes?
L1d cache:32K
L1i cache:32K
L2 cache: 256K
L3 cache: 3072K (lscpu)
MemTotal: 3936080 kB (meminfo)

3. Clock speed, the MIPS and the MFLOPS rating for your CPU?
Max Speed: 2600 Mhz (dmidecode)
BogoMIPS: 5187.88 (Bogus MIPS in lscpu)
MFLOPS rating: 13.05 GFLOPS (based on whetstone benchmark by https://milkyway.cs.rpi.edu/milkyway/cpu_list.php project)

4. Disk seek, latency, and transfer times (your local hard disk)?
Disk seek: 55 seek/seconds (ioping)
latency: 18.2 ms (ioping)
transfer: 221437 bytes/s (ioping)

5. How to know if your processor is 64b or 32b? How do you know if the processor supports ”hyperthreading”?
Output of CPU op mode in lscpu and Threads per core

6. OS version? Is 32 or 64 bit?
64 bit (lscpu)

7. Available VM size for user processes. Can it be changed?
SwapTotal: 4882428 kB
VmallocTotal: 34359738367 kB (meminfo)

Yes by making more swap space using dd command but data is lost

8. Limits on the stack space, heap space and static area. Can they be changed?
Stack space: 8192 (ulimit)
Heap space: unlimited (ulimit)

These can be changed using the ulimit command

9. What is a memory leak and how can you detect one?
Memory leak is the unnecessary usage of memory in heap allocated by a program which exited without freeing this memory.

It can be detect by frequent observation of memory using free and ps command like outputs

10. How can one cause a signal on stack overflow?
One can monitor the stack using getrlimit() system call and can initiate a signal (maybe SIGILL) when stack is critically low.

11. How can one tell how many page faults a process had?
ru_minflt of getrusage()

12. How can one tell how much user, system and elapsed time a process used.
Calling the executable with time

13. How can one time a procedure in a program? How accurate is this?
The CPU time of a procedure can be measured by noting the value of clock() before and after the call.

The real time of a procedure can be measured by doing the same with any function returning the real time.

Clock() cannot give the exact time for multithreaded programs as they use more cpu time in less wall clock time and the other wall clock time functions fail to take in account other processes.