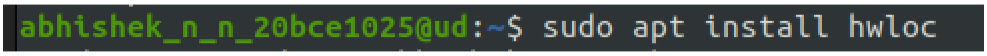
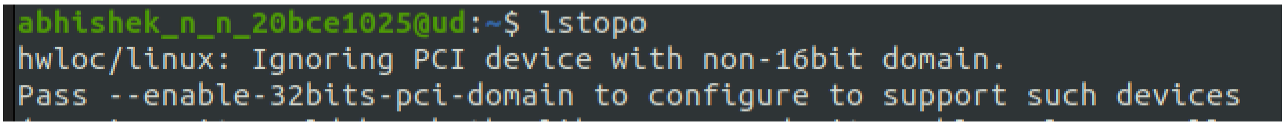
Name	:	Abhishek N N
Reg.no	:	20BCE1025
Email	:	abhishek.nn2020@vitstudent.ac.in
Mob	:	9482992712

CSE4001 Parallel and Distributed Computing

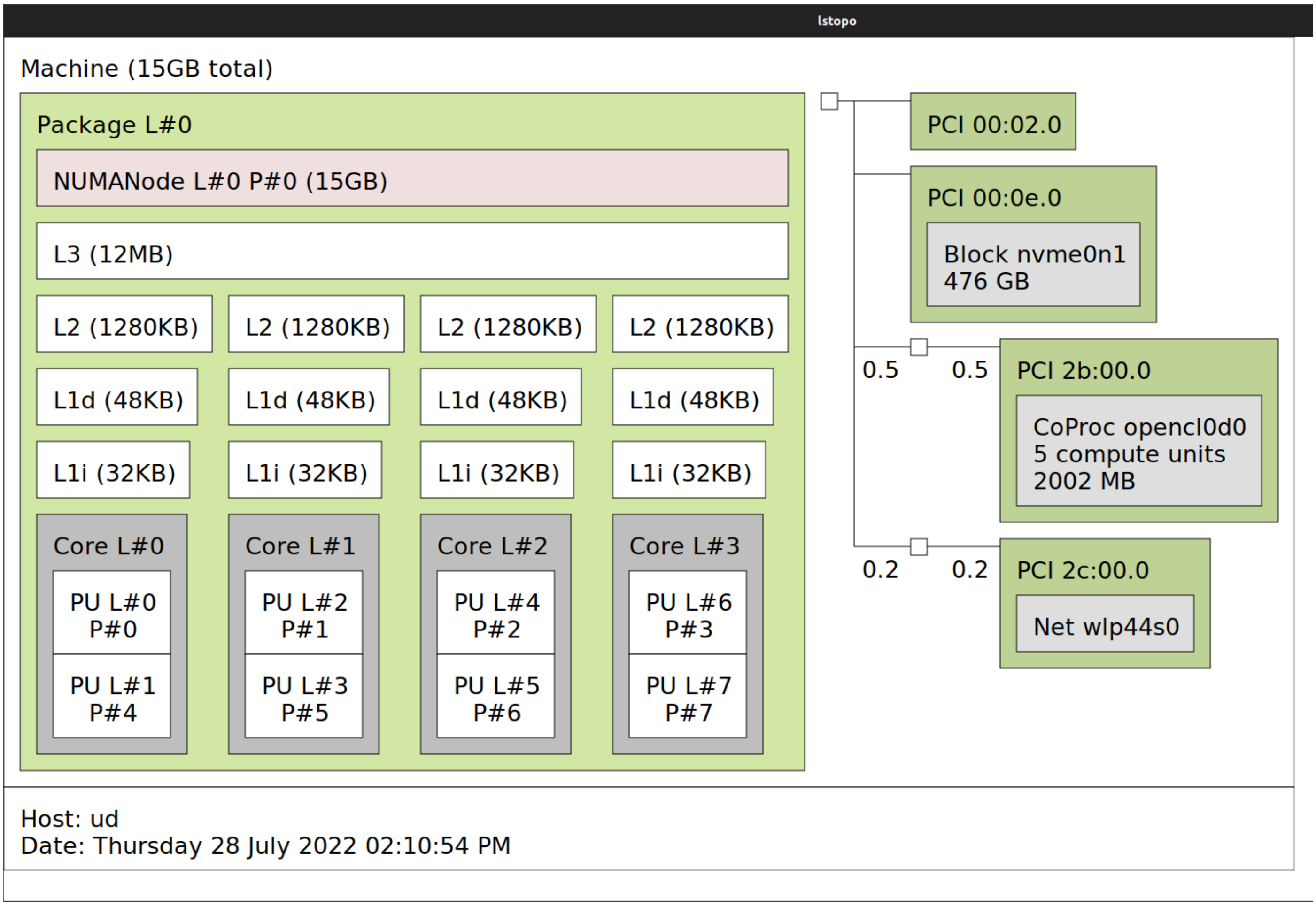
1. Display the processors layout of your system



```
abhishek_n_n_20bce1025@ud:~$ sudo apt install hwloc
```



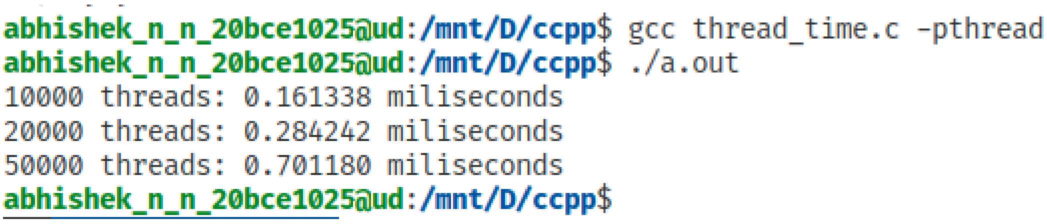
```
abhishek_n_n_20bce1025@ud:~$ lstopo
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
```



Host: ud
Date: Thursday 28 July 2022 02:10:54 PM

2. Write a multithreaded-thread program in **c** to create 10k, 20k, and 50k threads and measure the time taken for each thread group.

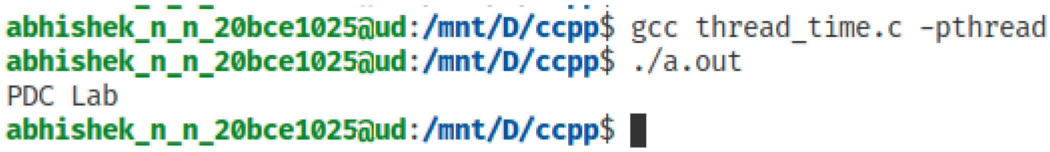
```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void *thread_function(void *arg) {}

int main() {
  int no[3] = {10000, 20000, 50000};

  for (int n = 0; n < 3; n++) {
    clock_t begin = clock();
    pthread_t threads[no[n]];
    int rc, i;

    for (i = 0; i < no[n]; i++) {
      rc = pthread_create(&threads[i], NULL, thread_function, NULL);

      if (rc) {
        printf("Error:unable to create thread, %d\n", rc);
        exit(-1);
      }
      pthread_join(threads[i], NULL);
    }
    clock_t end = clock();
    double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
    printf("%d threads: %f miliseconds\n", no[n], time_spent);
  }
}
```



```
abhishek_n_n_20bce1025@ud:/mnt/D/ccpp$ gcc thread_time.c -pthread
abhishek_n_n_20bce1025@ud:/mnt/D/ccpp$ ./a.out
10000 threads: 0.161338 miliseconds
20000 threads: 0.284242 miliseconds
50000 threads: 0.701180 miliseconds
abhishek_n_n_20bce1025@ud:/mnt/D/ccpp$
```

3. Write a program to create two threads. Thread1 has to print the print String1 “PDC” and Thread2 has to print the String2 “Lab”.
   Hint: Implement the message_function () is used as the start routine for the threads used it accepts a void pointer.
   Sample output:
   Thread1 prints: PDC
   Thread2 prints: Lab

```
#include <pthread.h>
#include <stdio.h>

void *message_function(void *arg) {
     char *msg = (char *)arg;
     printf("%s", msg);
}

int main() {
     pthread_t thread1, thread2;
     char *msg1 = "PDC", *msg2 = " Lab\n";
     pthread_create(&thread1, NULL, message_function, (void *)msg1);
     pthread_join(thread1, NULL);
     pthread_create(&thread2, NULL, message_function, (void *)msg2);
     pthread_join(thread2, NULL);
     return 0;
}
```



```
abhishek_n_n_20bce1025@ud:/mnt/D/ccpp$ gcc thread_time.c -pthread
abhishek_n_n_20bce1025@ud:/mnt/D/ccpp$ ./a.out
PDC Lab
abhishek_n_n_20bce1025@ud:/mnt/D/ccpp$ 
```