

20BCE1550

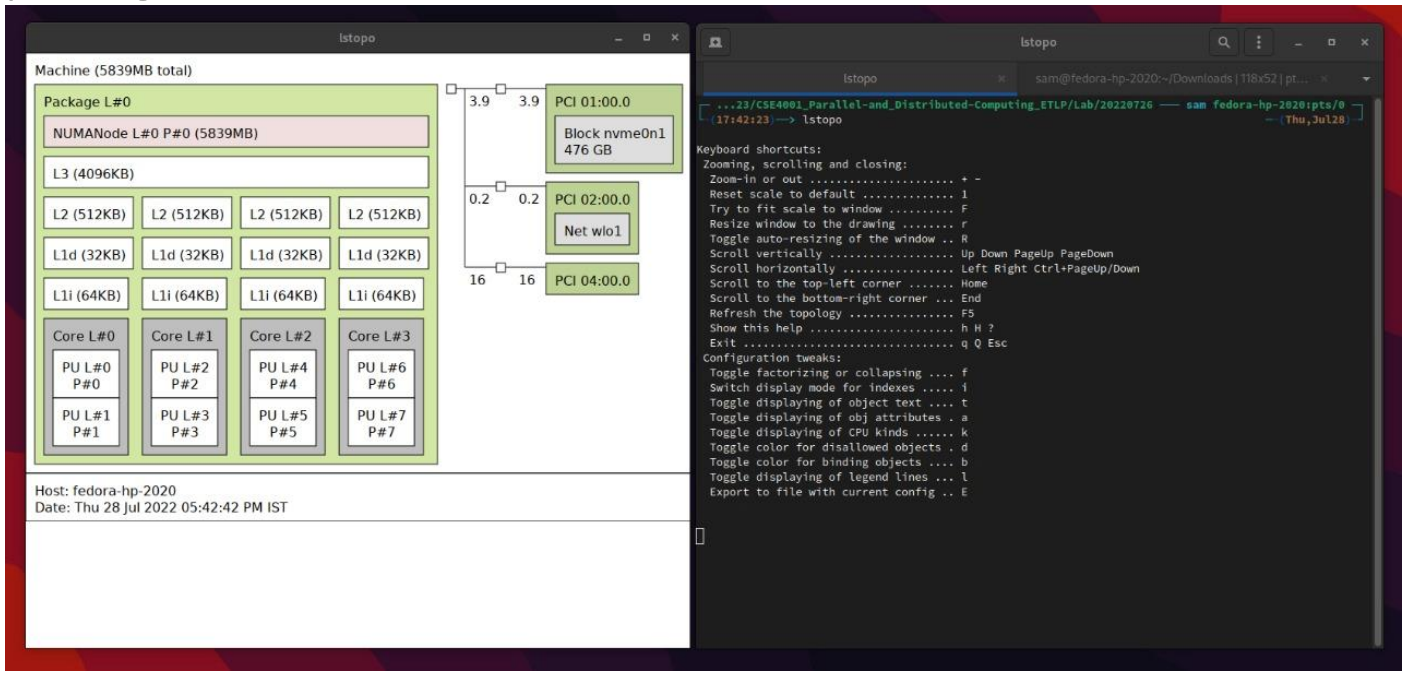
Samridh Anand Paatni

CSE4001 Lab 01

PThreads

Q1. Display the processors layout of your system.

The output for the command **lstopo**, after installing the hwloc package:



Q2. Write a multithreaded program in C to create 10k, 20k and 50k threads and measure the time taken for each thread group.

C Code:

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

void * void_function(void *message) {}

int main(int argc, char * argv[]) {
    pthread_t * threads;
    int num_threads = atoi(argv[1]); // because the cli argument is an ASCII code

    threads = (pthread_t *) calloc(num_threads, sizeof(pthread_t));

    clock_t t = clock();

    for (int i = 0; i < num_threads; i++) {
        pthread_create(&threads[i], NULL, void_function, NULL);
    }

    for (int i = 0; i < num_threads; i++) {
        pthread_join(threads[i], NULL);
    }

    t = clock() - t;

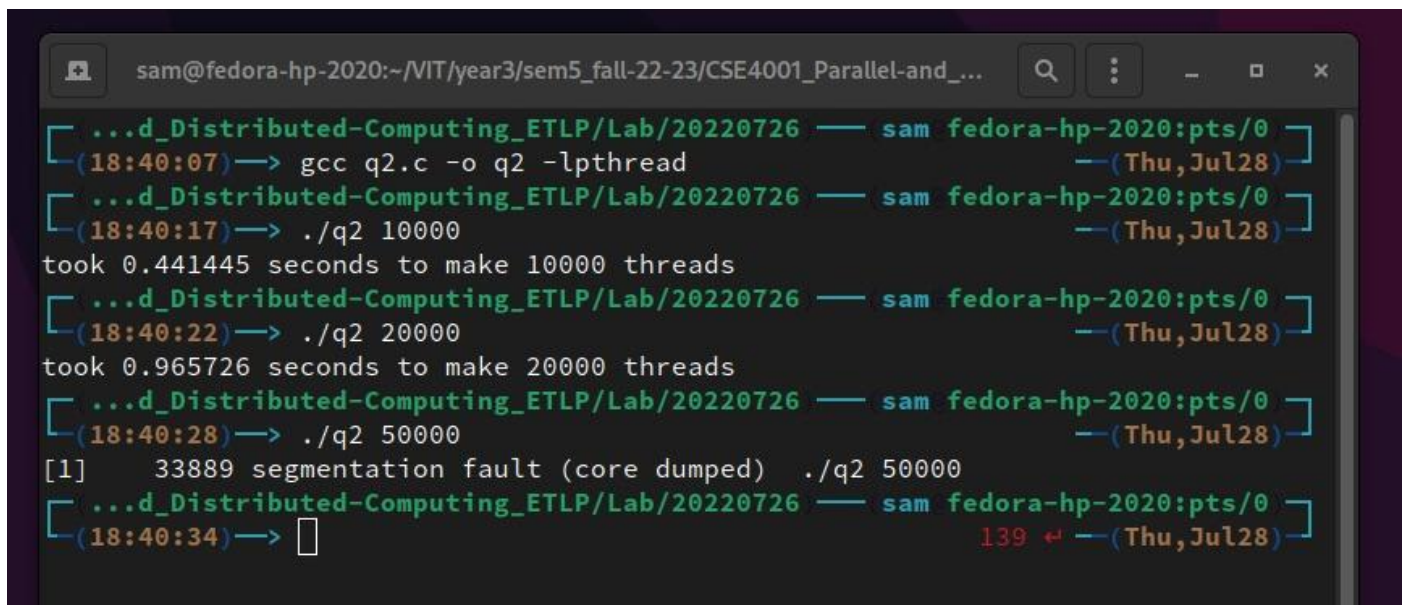
    printf(
        "took %f seconds to make %d threads\n",
        ((double) t)/CLOCKS_PER_SEC,
        num_threads
    );

    free(threads);

    return 0;
}
```

Output:

Creating 5000 pthreads resulted in a segmentation fault.



```
sam@fedora-hp-2020:~/VIT/year3/sem5_fall-22-23/CSE4001_Parallel-and_...  
[...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0]  
(18:40:07)→ gcc q2.c -o q2 -lpthread —(Thu,Jul28)  
[...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0]  
(18:40:17)→ ./q2 10000 —(Thu,Jul28)  
took 0.441445 seconds to make 10000 threads  
[...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0]  
(18:40:22)→ ./q2 20000 —(Thu,Jul28)  
took 0.965726 seconds to make 20000 threads  
[...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0]  
(18:40:28)→ ./q2 50000 —(Thu,Jul28)  
[1] 33889 segmentation fault (core dumped) ./q2 50000  
[...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0]  
(18:40:34)→ 139 ← —(Thu,Jul28)
```

Q3. Write a program to create 2 threads. Thread 1 has to print “PDC” and thread 2 has to print “lab”.

C Code:

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

void * message_function(void *message) {
    printf("%s\n", ((char *) message));
}

int main() {
    pthread_t t1, t2;

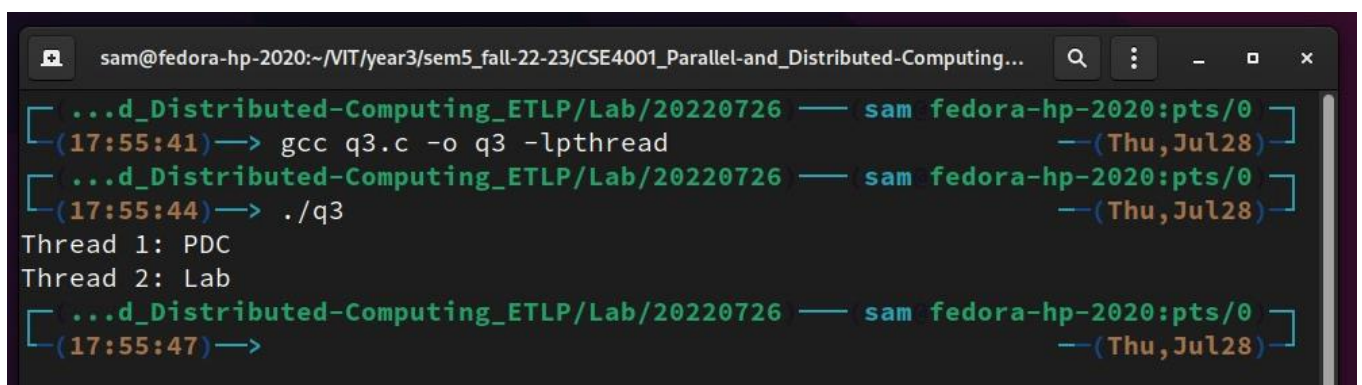
    char *m1 = "Thread 1: PDC";
    char *m2 = "Thread 2: Lab";

    pthread_create(&t1, NULL, message_function, (void *) m1);
    pthread_create(&t2, NULL, message_function, (void *) m2);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    return 0;
}
```

Output:



```
sam@fedora-hp-2020:~/VIT/year3/sem5_fall-22-23/CSE4001_Parallel-and_Distributed-Computing...
[ ...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0 ]
(17:55:41)→ gcc q3.c -o q3 -lpthread —(Thu, Jul28)
[ ...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0 ]
(17:55:44)→ ./q3 —(Thu, Jul28)
Thread 1: PDC
Thread 2: Lab
[ ...d_Distributed-Computing_ETLP/Lab/20220726 — sam fedora-hp-2020:pts/0 ]
(17:55:47)→ —(Thu, Jul28)
```