

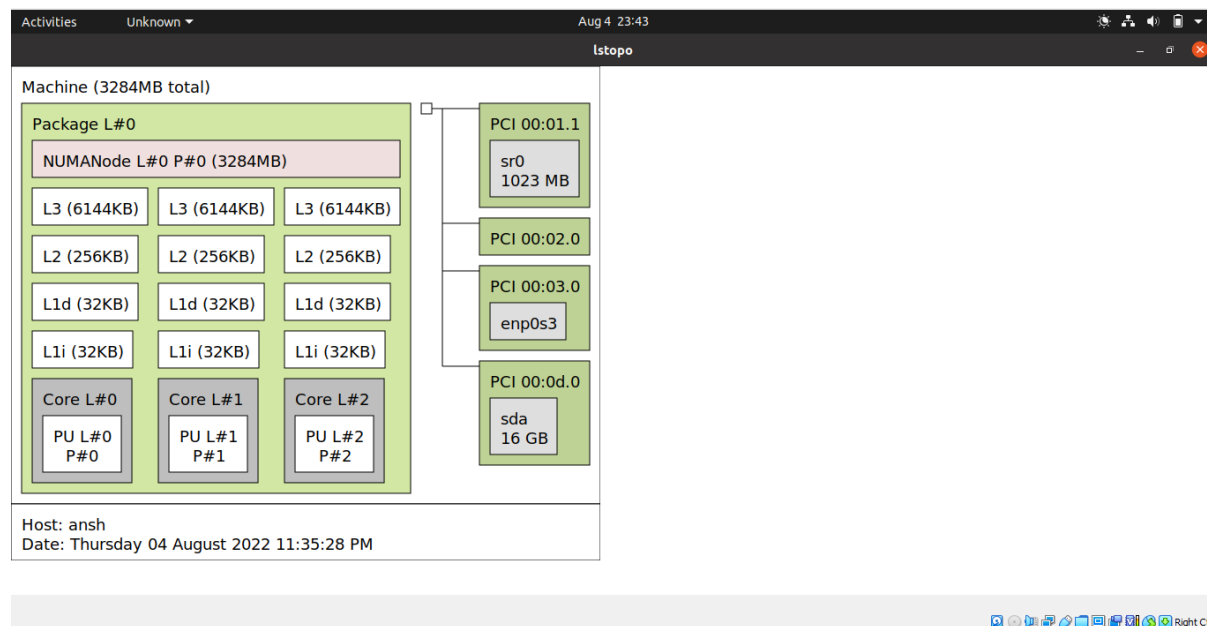
NAME – ANSH GOEL

REGISTER NO.- 20BCE1798

PARALLEL AND DISTRIBUTED COMPUTING

LAB – 1

1. Display the processors layout of your system



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

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <time.h>
void *myThreadFun(void *vargp)
{
    int *myid=(int *)vargp;
    // printf("Thread Number: %d\n",*myid);

}
int main()
{
    pthread_t thread_id;
```

```

int i;
printf("Before Thread\n");
clock_t t; t = clock();
for(i=0;i<1000;i++)
pthread_create(&thread_id, NULL, myThreadFun, (void *)&i);
t=clock()-t;

double time_taken = ((double)t)/CLOCKS_PER_SEC;

t = clock();

for(i=0;i<10000;i++)
pthread_create(&thread_id, NULL, myThreadFun, (void *)&i);
t=clock()-t;time_taken = ((double)t)/CLOCKS_PER_SEC;
printf("10k thread took %f seconds to execute \n", time_taken);
t = clock();
for(i=0;i<20000;i++)
pthread_create(&thread_id, NULL, myThreadFun, (void *)&i);
t=clock()-t;time_taken = ((double)t)/CLOCKS_PER_SEC;
printf("20k thread took %f seconds to execute \n", time_taken);

t = clock(); for(i=0;i<50000;i++)
pthread_create(&thread_id, NULL, myThreadFun, (void *)&i);
t=clock()-t;time_taken = ((double)t)/CLOCKS_PER_SEC;
printf("50k thread took %f seconds to execute \n", time_taken);
pthread_exit(NULL);

return 0;

}

```

```

ansh@ansh:~$ gedit pdclab1.c
ansh@ansh:~$ gcc pdclab1.c -o pdclab1 -lpthread
ansh@ansh:~$ ./pdclab1
Before Thread
10k thread took 0.528162 seconds to execute
20k thread took 1.135495 seconds to execute
50k thread took 0.226291 seconds to execute
libgcc_s.so.1 must be installed for pthread_cancel to work
Aborted (core dumped)
ansh@ansh:~$

```

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

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
void *Thread1(void *vargp)
{
    int *myid=(int *)vargp;
    printf("Thread 1 : PDC\n");
}

void *Thread2(void *vargp)
{ int *myid=(int *)vargp;
  printf("Thread 2 : Lab\n");
}

int main()
{
    pthread_t thread_id;
    int i=1;
    pthread_create(&thread_id, NULL, Thread1, (void *)&i);
    i=2;
    pthread_create(&thread_id, NULL, Thread2, (void *)&i);
    pthread_exit(NULL);

    return 0;
}
```

```
ansh@ansh:~$ gedit pdclab2.c
ansh@ansh:~$ gcc pdclab2.c -o pdclab2 -lpthread
ansh@ansh:~$ ./pdclab2
Thread 2 : Lab
Thread 1 : PDC
ansh@ansh:~$
```

