Name: *Chillara V L N S Pavana Vamsi*

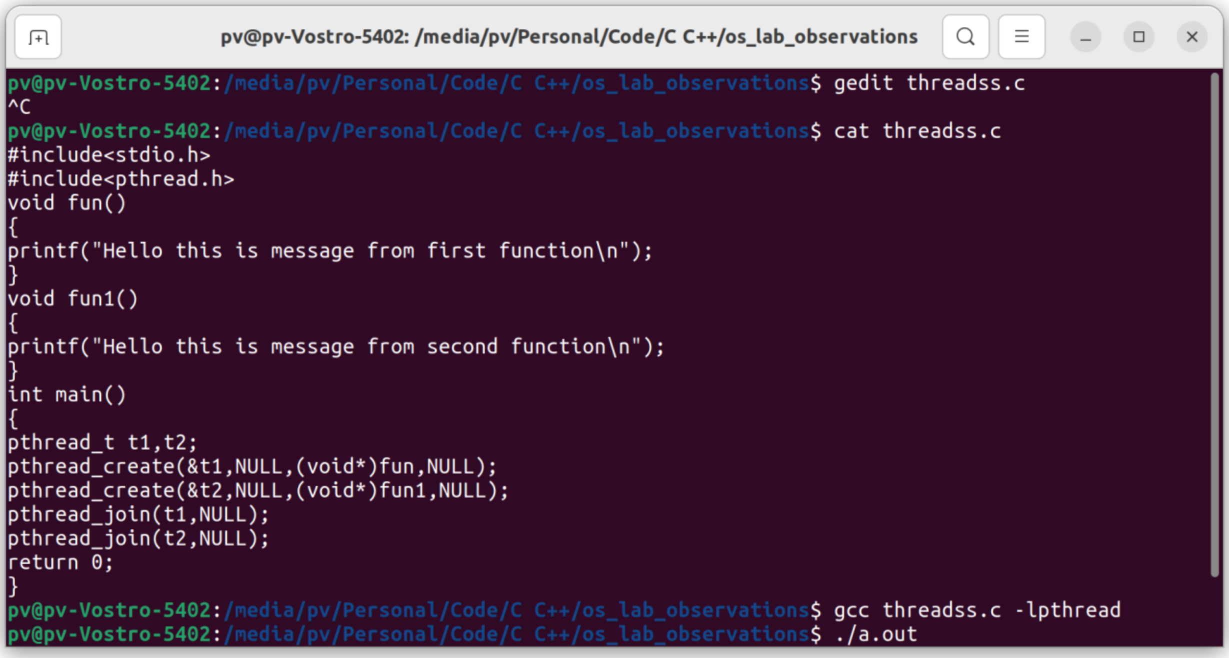
Reg.no: *21BCE5095*

Date: *12/1/2023*

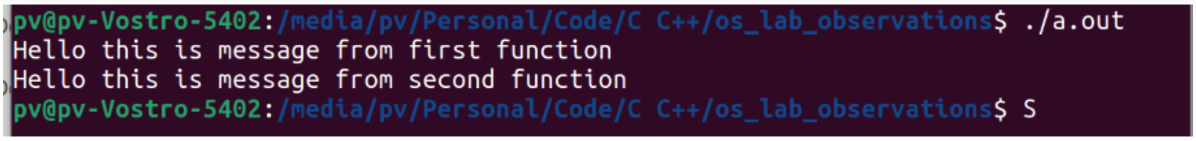
Faculty: *M Sivagami*

L21_L22_Lab4 Exercises

1. Design two threads to display two different messages using two different functions.

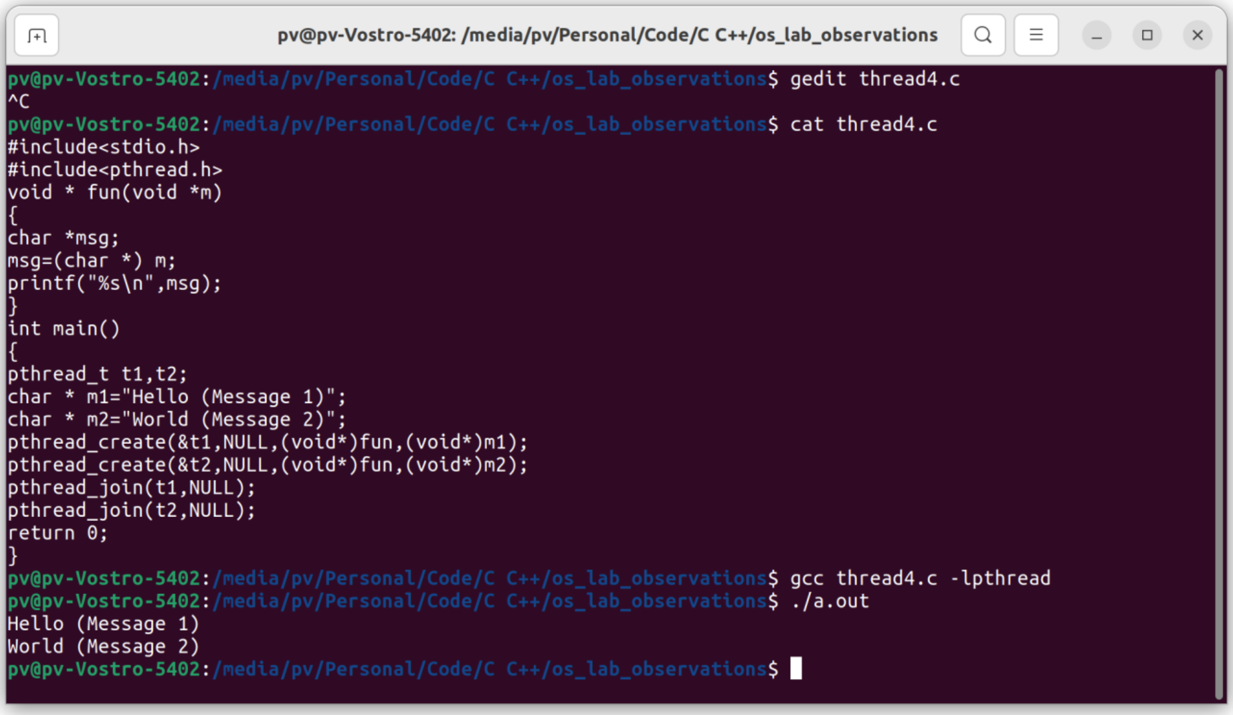

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit threadss.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat threadss.c
#include<stdio.h>
#include<pthread.h>
void fun()
{
printf("Hello this is message from first function\n");
}
void fun1()
{
printf("Hello this is message from second function\n");
}
int main()
{
pthread_t t1,t2;
pthread_create(&t1,NULL,(void*)fun,NULL);
pthread_create(&t2,NULL,(void*)fun1,NULL);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc threadss.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
```



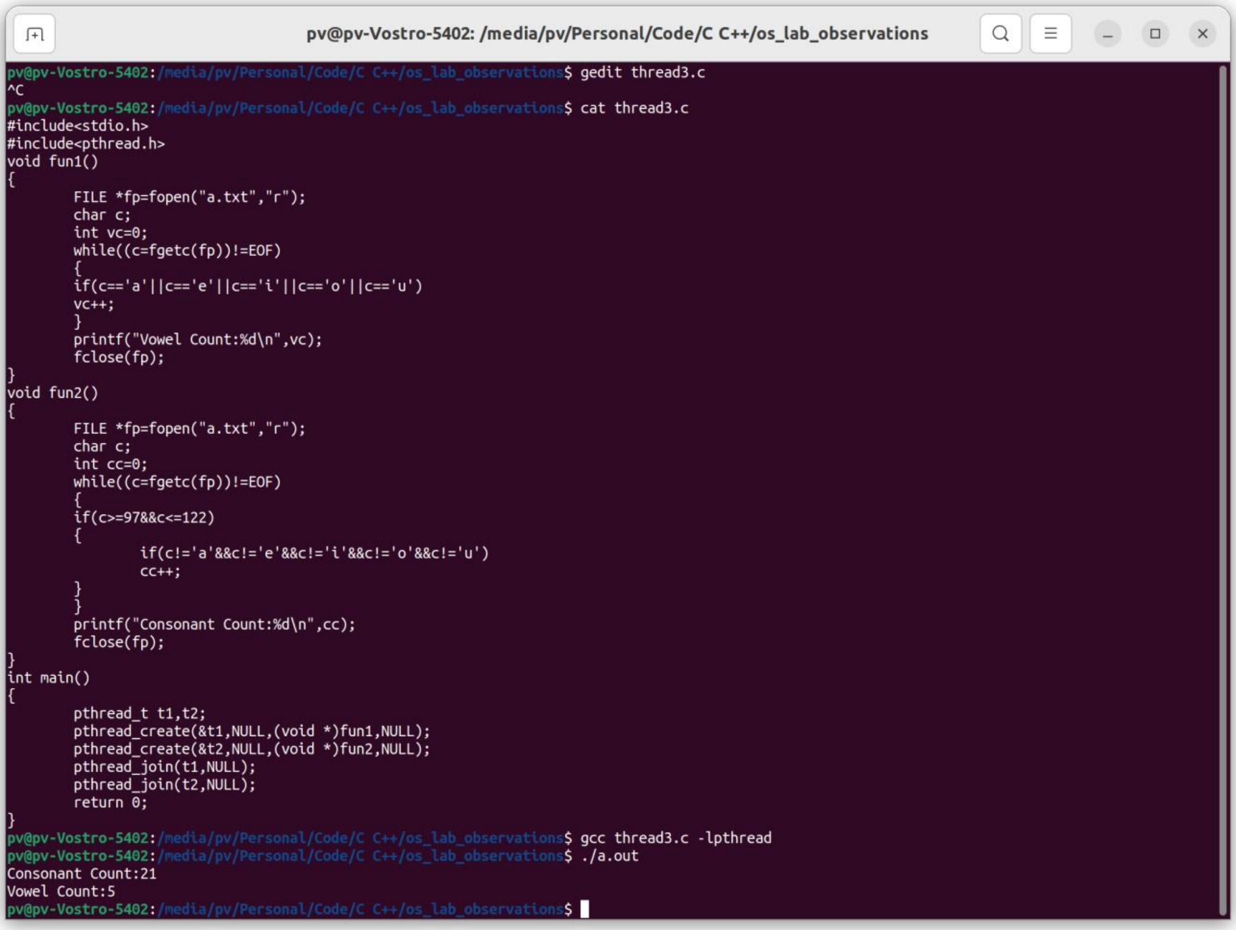
```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Hello this is message from first function
Hello this is message from second function
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ S
```

2. Design two threads to display two different messages using single function.


pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

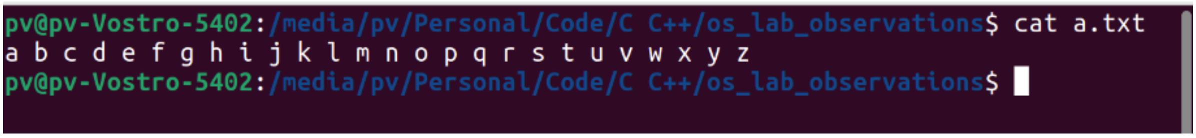
```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit thread4.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat thread4.c
#include<stdio.h>
#include<pthread.h>
void * fun(void *m)
{
char *msg;
msg=(char *) m;
printf("%s\n",msg);
}
int main()
{
pthread_t t1,t2;
char * m1="Hello (Message 1)";
char * m2="World (Message 2)";
pthread_create(&t1,NULL,(void*)fun,(void*)m1);
pthread_create(&t2,NULL,(void*)fun,(void*)m2);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc thread4.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Hello (Message 1)
World (Message 2)
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```

3. Design two threads to count the vowels and consonants either from "a.txt" file or a given string.



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit thread3.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat thread3.c
#include<stdio.h>
#include<pthread.h>
void fun1()
{
        FILE *fp=fopen("a.txt","r");
        char c;
        int vc=0;
        while((c=fgetc(fp))!=EOF)
        {
        if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u')
        vc++;
        }
        printf("Vowel Count:%d\n",vc);
        fclose(fp);
}
void fun2()
{
        FILE *fp=fopen("a.txt","r");
        char c;
        int cc=0;
        while((c=fgetc(fp))!=EOF)
        {
        if(c>=97&&c<=122)
        {
                if(c!='a'&&c!='e'&&c!='i'&&c!='o'&&c!='u')
                cc++;
        }
        }
        printf("Consonant Count:%d\n",cc);
        fclose(fp);
}
int main()
{
        pthread_t t1,t2;
        pthread_create(&t1,NULL,(void *)fun1,NULL);
        pthread_create(&t2,NULL,(void *)fun2,NULL);
        pthread_join(t1,NULL);
        pthread_join(t2,NULL);
        return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc thread3.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Consonant Count:21
Vowel Count:5
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```

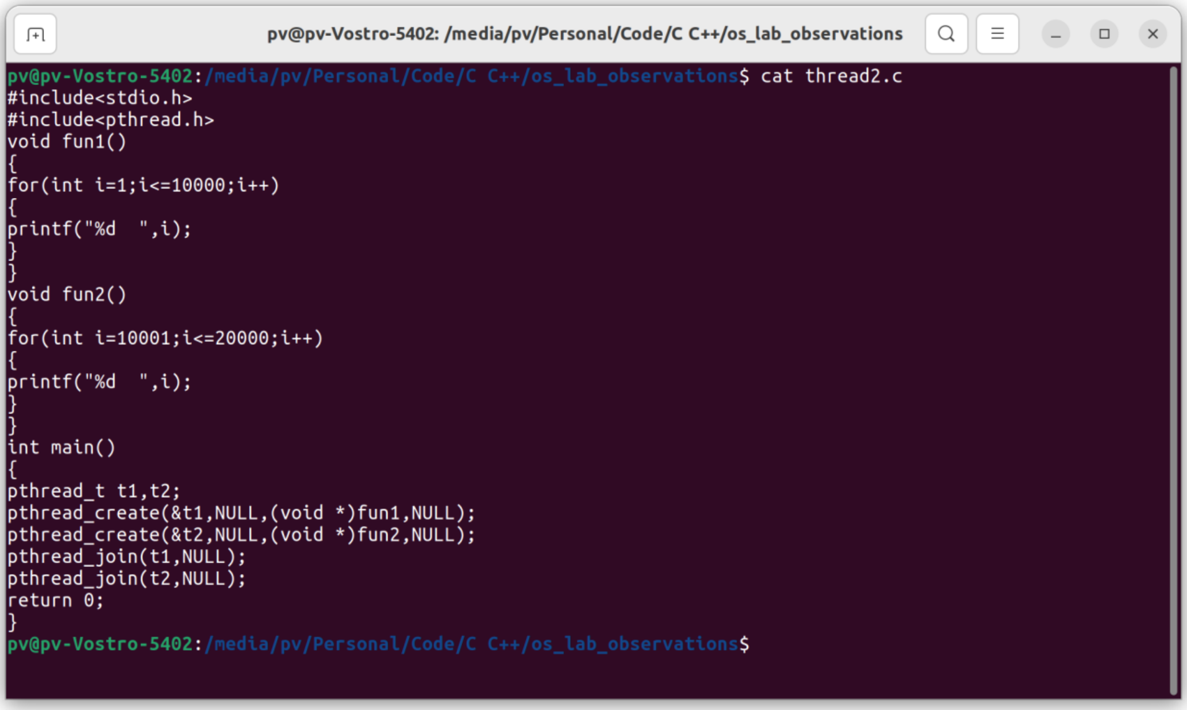


```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat a.txt
a b c d e f g h i j k l m n o p q r s t u v w x y z
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```

4. Design two threads to display the numbers (i) from 1 to 10000 (ii) from 10001 to 20000 using (a) two different functions or (ii) single function. Discuss the nature of the output generated by the two threads in command prompt.

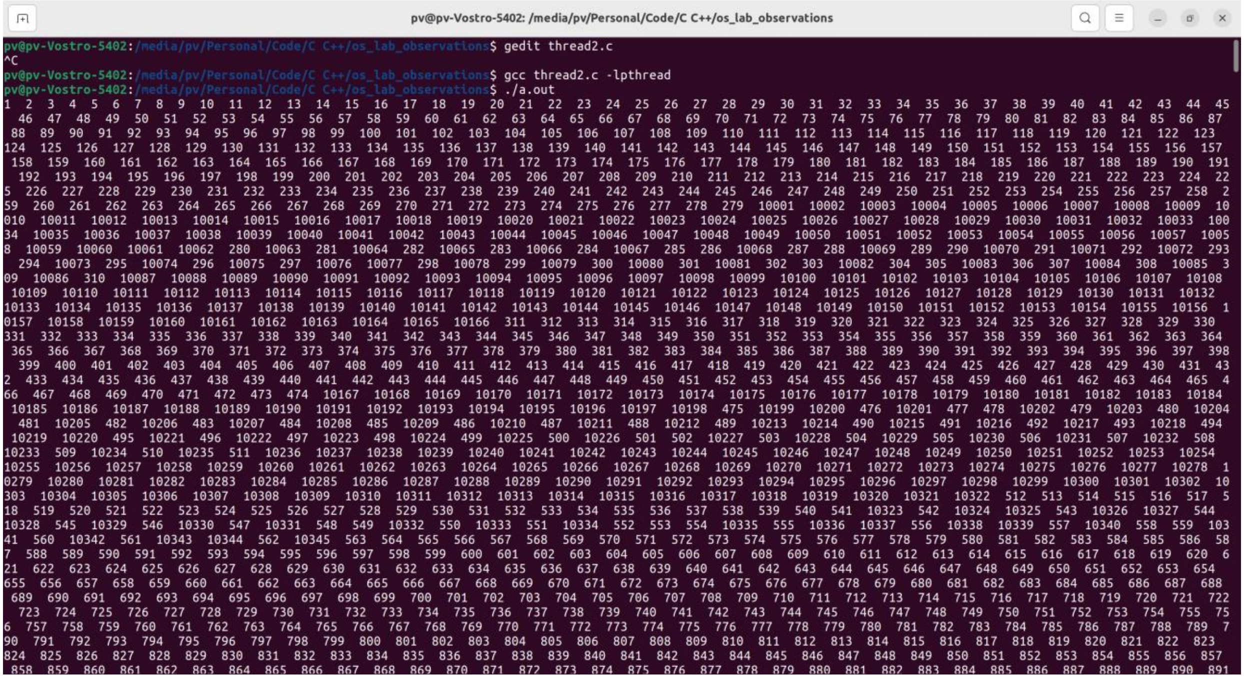
- **Two different functions:**

Here two threads functions run simultaneously so the numbers come in mix of both 1st function and 2nd function.



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat thread2.c
#include<stdio.h>
#include<pthread.h>
void fun1()
{
for(int i=1;i<=10000;i++)
{
printf("%d  ",i);
}
}
void fun2()
{
for(int i=10001;i<=20000;i++)
{
printf("%d  ",i);
}
}
int main()
{
pthread_t t1,t2;
pthread_create(&t1,NULL,(void *)fun1,NULL);
pthread_create(&t2,NULL,(void *)fun2,NULL);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$
```

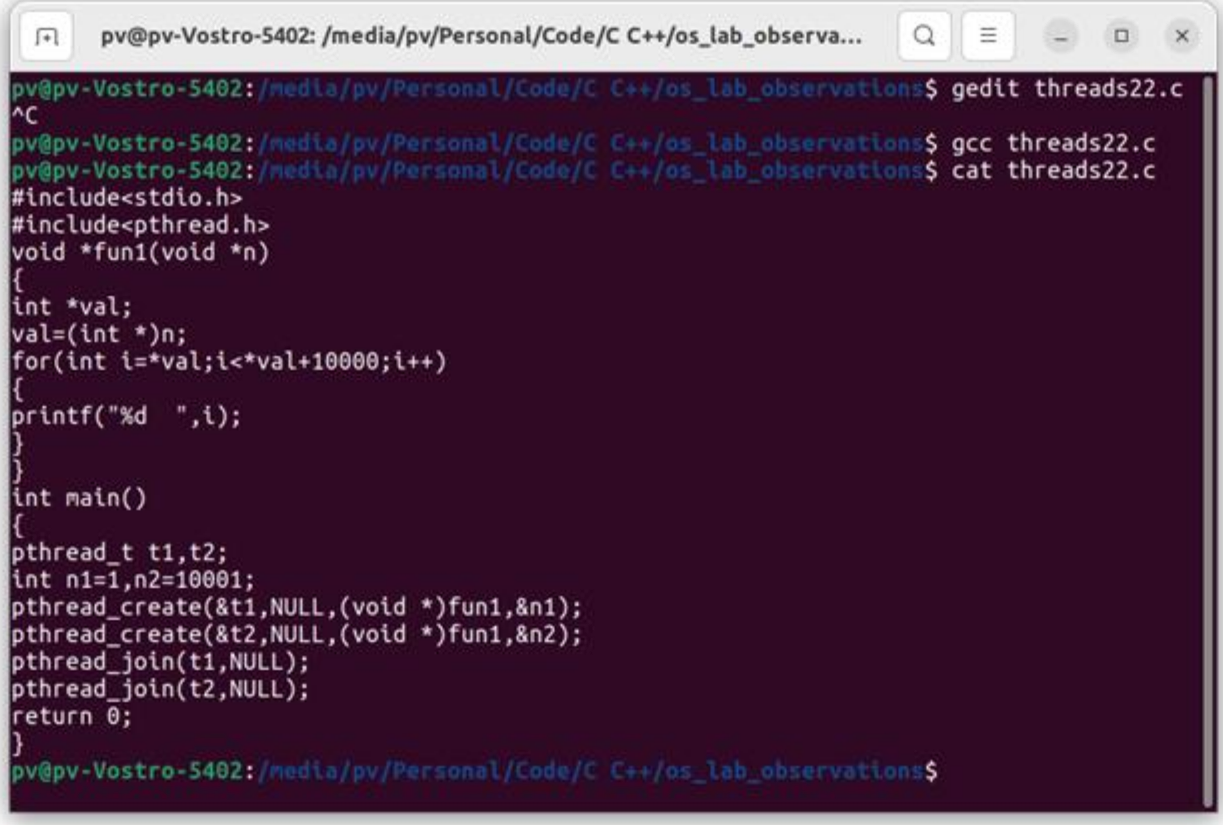


pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit thread2.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc thread2.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45
 46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87
 88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123
124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157
 158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191
  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  22
5  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  2
59  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  10001  10002  10003  10004  10005  10006  10007  10008  10009  10
010  10011  10012  10013  10014  10015  10016  10017  10018  10019  10020  10021  10022  10023  10024  10025  10026  10027  10028  10029  10030  10031  10032  10033  100
34  10035  10036  10037  10038  10039  10040  10041  10042  10043  10044  10045  10046  10047  10048  10049  10050  10051  10052  10053  10054  10055  10056  10057  1005
8  10059  10060  10061  10062  280  10063  281  10064  282  10065  283  10066  284  10067  285  286  10068  287  288  10069  289  290  10070  291  10071  292  10072  293
  294  10073  295  10074  296  10075  297  10076  10077  298  10078  299  10079  300  10080  301  10081  302  303  10082  304  305  10083  306  307  10084  308  10085  3
09  10086  310  10087  10088  10089  10090  10091  10092  10093  10094  10095  10096  10097  10098  10099  10100  10101  10102  10103  10104  10105  10106  10107  10108
 10109  10110  10111  10112  10113  10114  10115  10116  10117  10118  10119  10120  10121  10122  10123  10124  10125  10126  10127  10128  10129  10130  10131  10132
10133  10134  10135  10136  10137  10138  10139  10140  10141  10142  10143  10144  10145  10146  10147  10148  10149  10150  10151  10152  10153  10154  10155  10156  1
0157  10158  10159  10160  10161  10162  10163  10164  10165  10166  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330
331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364
 365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398
  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  43
2  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  4
66  467  468  469  470  471  472  473  474  10167  10168  10169  10170  10171  10172  10173  10174  10175  10176  10177  10178  10179  10180  10181  10182  10183  10184
 10185  10186  10187  10188  10189  10190  10191  10192  10193  10194  10195  10196  10197  10198  475  10199  10200  476  10201  477  478  10202  479  10203  480  10204
  481  10205  482  10206  483  10207  484  10208  485  10209  486  10210  487  10211  488  10212  489  10213  10214  490  10215  491  10216  492  10217  493  10218  494
 10219  10220  495  10221  496  10222  497  10223  498  10224  499  10225  500  10226  501  502  10227  503  10228  504  10229  505  10230  506  10231  507  10232  508
10233  509  10234  510  10235  511  10236  10237  10238  10239  10240  10241  10242  10243  10244  10245  10246  10247  10248  10249  10250  10251  10252  10253  10254
10255  10256  10257  10258  10259  10260  10261  10262  10263  10264  10265  10266  10267  10268  10269  10270  10271  10272  10273  10274  10275  10276  10277  10278  1
0279  10280  10281  10282  10283  10284  10285  10286  10287  10288  10289  10290  10291  10292  10293  10294  10295  10296  10297  10298  10299  10300  10301  10302  10
303  10304  10305  10306  10307  10308  10309  10310  10311  10312  10313  10314  10315  10316  10317  10318  10319  10320  10321  10322  512  513  514  515  516  517  5
18  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  10323  542  10324  10325  543  10326  10327  544
10328  545  10329  546  10330  547  10331  548  549  10332  550  10333  551  10334  552  553  554  10335  555  10336  10337  556  10338  10339  557  10340  558  559  103
41  560  10342  561  10343  10344  562  10345  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  58
7  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  6
21  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654
655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688
 689  690  691  692  693  694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722
  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742  743  744  745  746  747  748  749  750  751  752  753  754  755  75
6  757  758  759  760  761  762  763  764  765  766  767  768  769  770  771  772  773  774  775  776  777  778  779  780  781  782  783  784  785  786  787  788  789  7
90  791  792  793  794  795  796  797  798  799  800  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816  817  818  819  820  821  822  823
824  825  826  827  828  829  830  831  832  833  834  835  836  837  838  839  840  841  842  843  844  845  846  847  848  849  850  851  852  853  854  855  856  857
 858  859  860  861  862  863  864  865  866  867  868  869  870  871  872  873  874  875  876  877  878  879  880  881  882  883  884  885  886  887  888  889  890  891
```

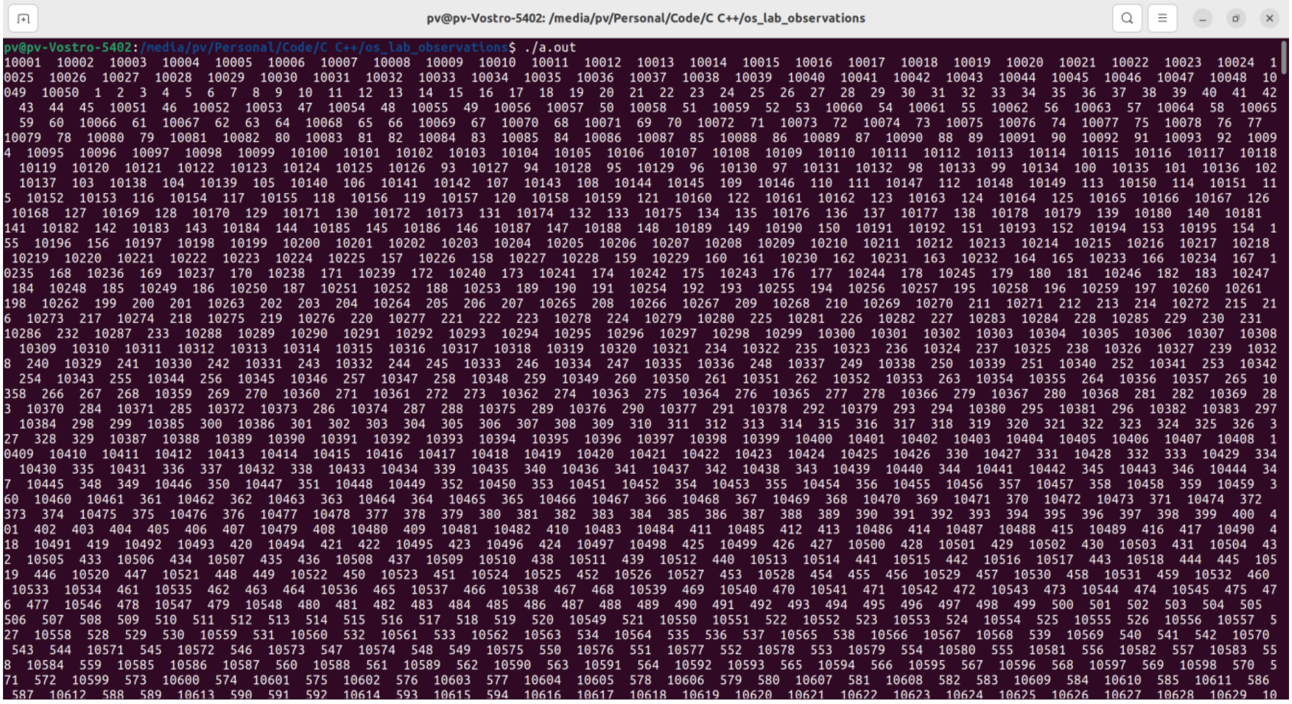
Single function:

Here both threads are running simultaneously in that process. So, the numbers print in zig-zag order.



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observa...

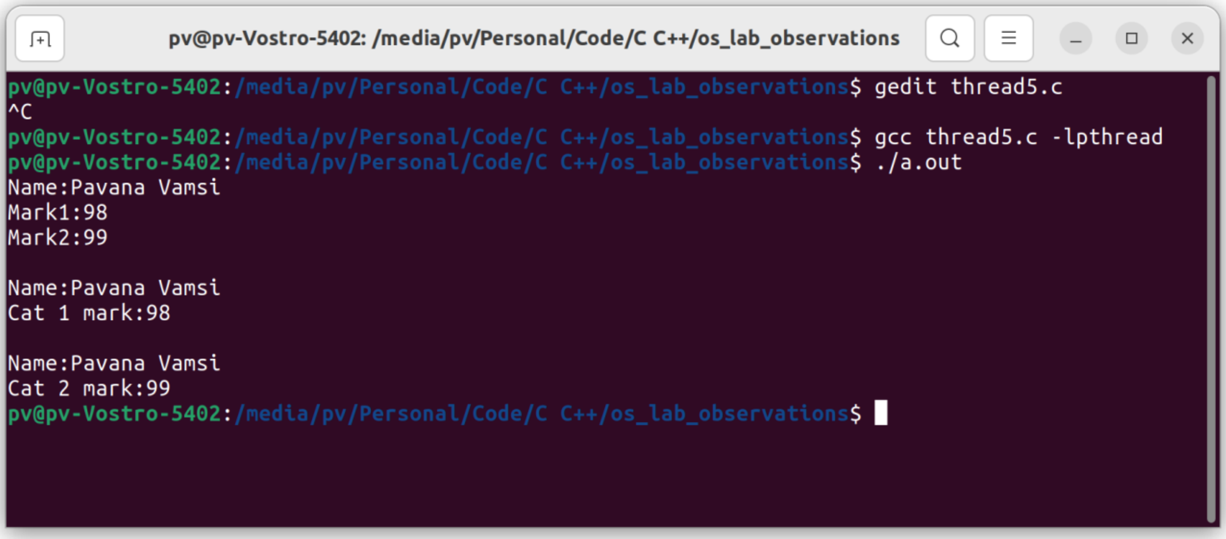
```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit threads22.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc threads22.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat threads22.c
#include<stdio.h>
#include<pthread.h>
void *fun1(void *n)
{
int *val;
val=(int *)n;
for(int i=*val;i<*val+10000;i++)
{
printf("%d  ",i);
}
}
int main()
{
pthread_t t1,t2;
int n1=1,n2=10001;
pthread_create(&t1,NULL,(void *)fun1,&n1);
pthread_create(&t2,NULL,(void *)fun1,&n2);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$
```



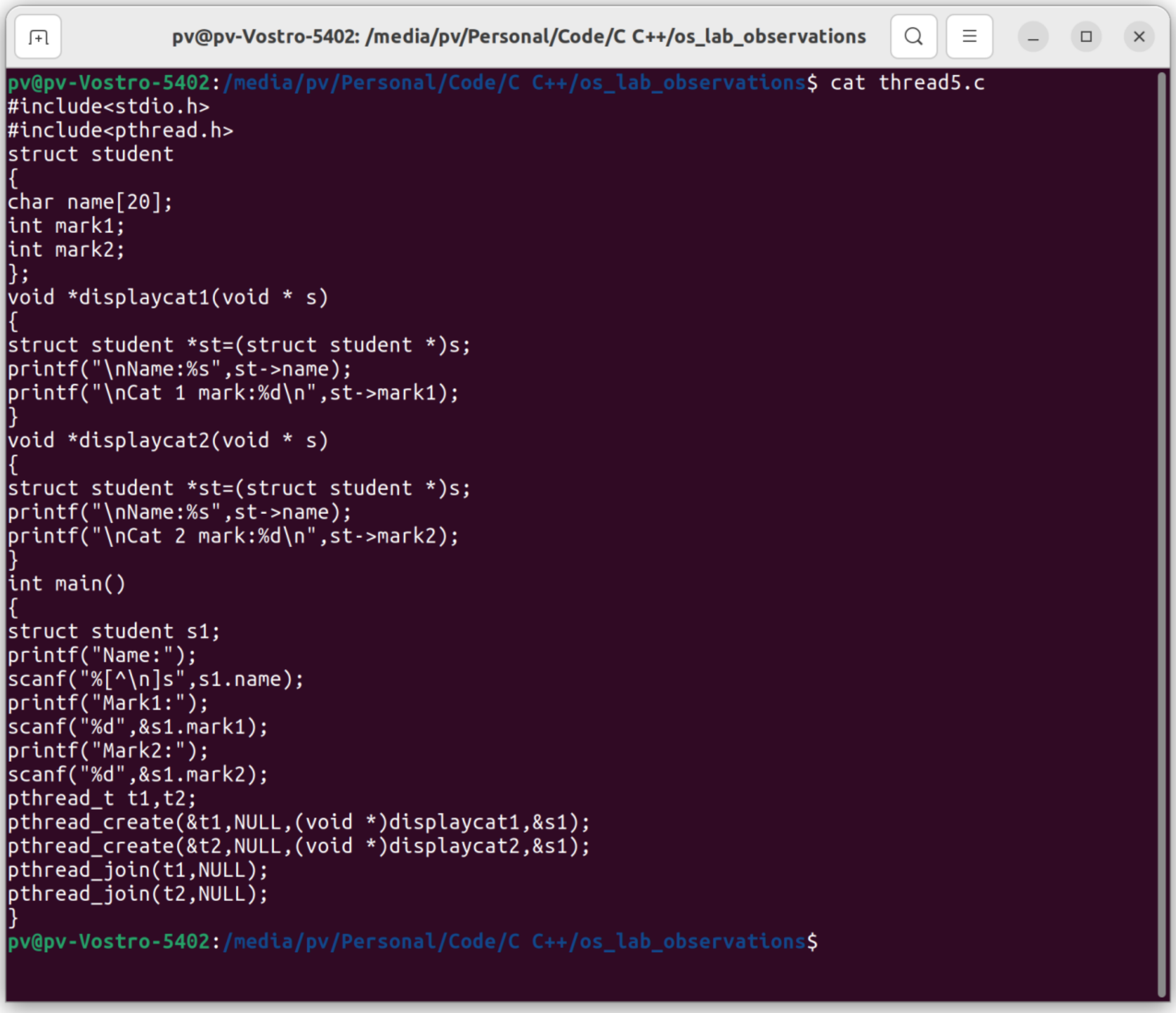
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
10001  10002  10003  10004  10005  10006  10007  10008  10009  10010  10011  10012  10013  10014  10015  10016  10017  10018  10019  10020  10021  10022  10023  10024  1
0025  10026  10027  10028  10029  10030  10031  10032  10033  10034  10035  10036  10037  10038  10039  10040  10041  10042  10043  10044  10045  10046  10047  10048  10
049  10050  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42
  43  44  45  10051  46  10052  10053  47  10054  48  10055  49  10056  10057  50  10058  51  10059  52  53  10060  54  10061  55  10062  56  10063  57  10064  58  10065
  59  60  10066  61  10067  62  63  64  10068  65  66  10069  67  10070  68  10071  69  70  10072  71  10073  72  10074  73  10075  10076  74  10077  75  10078  76  77
10079  78  10080  79  10081  10082  80  10083  81  82  10084  83  10085  84  10086  10087  85  10088  86  10089  87  10090  88  89  10091  90  10092  91  10093  92  1009
4  10095  10096  10097  10098  10099  10100  10101  10102  10103  10104  10105  10106  10107  10108  10109  10110  10111  10112  10113  10114  10115  10116  10117  10118
  10119  10120  10121  10122  10123  10124  10125  10126  93  10127  94  10128  95  10129  96  10130  97  10131  10132  98  10133  99  10134  100  10135  101  10136  102
  10137  103  10138  104  10139  105  10140  106  10141  10142  107  10143  108  10144  10145  109  10146  110  111  10147  112  10148  10149  113  10150  114  10151  11
5  10152  10153  116  10154  117  10155  118  10156  119  10157  120  10158  10159  121  10160  122  10161  10162  123  10163  124  10164  125  10165  10166  10167  126
 10168  127  10169  128  10170  129  10171  130  10172  10173  131  10174  132  133  10175  134  135  10176  136  137  10177  138  10178  10179  139  10180  140  10181
141  10182  142  10183  143  10184  144  10185  145  10186  146  10187  147  10188  148  10189  149  10190  150  10191  10192  151  10193  152  10194  153  10195  154  1
55  10196  156  10197  10198  10199  10200  10201  10202  10203  10204  10205  10206  10207  10208  10209  10210  10211  10212  10213  10214  10215  10216  10217  10218
 10219  10220  10221  10222  10223  10224  10225  157  10226  158  10227  10228  159  10229  160  161  10230  162  10231  163  10232  164  165  10233  166  10234  167  1
0235  168  10236  169  10237  170  10238  171  10239  172  10240  173  10241  174  10242  175  10243  176  177  10244  178  10245  179  180  181  10246  182  183  10247
 184  10248  185  10249  186  10250  187  10251  10252  188  10253  189  190  191  10254  192  193  10255  194  10256  10257  195  10258  196  10259  197  10260  10261
198  10262  199  200  201  10263  202  203  204  10264  205  206  207  10265  208  10266  10267  209  10268  210  10269  10270  211  10271  212  213  214  10272  215  21
6  10273  217  10274  218  10275  219  10276  220  10277  221  222  223  10278  224  10279  10280  225  10281  226  10282  227  10283  10284  228  10285  229  230  231
10286  232  10287  233  10288  10289  10290  10291  10292  10293  10294  10295  10296  10297  10298  10299  10300  10301  10302  10303  10304  10305  10306  10307  10308
  10309  10310  10311  10312  10313  10314  10315  10316  10317  10318  10319  10320  10321  234  10322  235  10323  236  10324  237  10325  238  10326  10327  239  1032
8  240  10329  241  10330  242  10331  243  10332  244  245  10333  246  10334  247  10335  10336  248  10337  249  10338  250  10339  251  10340  252  10341  253  10342
  254  10343  255  10344  256  10345  10346  257  10347  258  10348  259  10349  260  10350  261  10351  262  10352  10353  263  10354  10355  264  10356  10357  265  10
358  266  267  268  10359  269  270  10360  271  10361  272  273  10362  274  10363  275  10364  276  10365  277  278  10366  279  10367  280  10368  281  282  10369  28
3  10370  284  10371  285  10372  10373  286  10374  287  288  10375  289  10376  290  10377  291  10378  292  10379  293  294  10380  295  10381  296  10382  10383  297
  10384  298  299  10385  300  10386  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  3
27  328  329  10387  10388  10389  10390  10391  10392  10393  10394  10395  10396  10397  10398  10399  10400  10401  10402  10403  10404  10405  10406  10407  10408  1
0409  10410  10411  10412  10413  10414  10415  10416  10417  10418  10419  10420  10421  10422  10423  10424  10425  10426  330  10427  331  10428  332  333  10429  334
  10430  335  10431  336  337  10432  338  10433  10434  339  10435  340  10436  341  10437  342  10438  343  10439  10440  344  10441  10442  345  10443  346  10444  34
7  10445  348  349  10446  350  10447  351  10448  10449  352  10450  353  10451  10452  354  10453  355  10454  356  10455  10456  357  10457  358  10458  359  10459  3
60  10460  10461  361  10462  362  10463  363  10464  364  10465  365  10466  10467  366  10468  367  10469  368  10470  369  10471  370  10472  10473  371  10474  372
373  374  10475  375  10476  376  10477  10478  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  4
01  402  403  404  405  406  407  10479  408  10480  409  10481  10482  410  10483  10484  411  10485  412  413  10486  414  10487  10488  415  10489  416  417  10490  4
18  10491  419  10492  10493  420  10494  421  422  10495  423  10496  424  10497  10498  425  10499  426  427  10500  428  10501  429  10502  430  10503  431  10504  43
2  10505  433  10506  434  10507  435  436  10508  437  10509  10510  438  10511  439  10512  440  10513  10514  441  10515  442  10516  10517  443  10518  444  445  105
19  446  10520  447  10521  448  449  10522  450  10523  451  10524  10525  452  10526  10527  453  10528  454  455  456  10529  457  10530  458  10531  459  10532  460
 10533  10534  461  10535  462  463  464  10536  465  10537  466  10538  467  468  10539  469  10540  470  10541  471  10542  472  10543  473  10544  474  10545  475  47
6  477  10546  478  10547  479  10548  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505
506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  10549  521  10550  10551  522  10552  523  10553  524  10554  525  10555  526  10556  10557  5
27  10558  528  529  530  10559  531  10560  532  10561  533  10562  10563  534  10564  535  536  537  10565  538  10566  10567  10568  539  10569  540  541  542  10570
 543  544  10571  545  10572  546  10573  547  10574  548  549  10575  550  10576  551  10577  552  10578  553  10579  554  10580  555  10581  556  10582  557  10583  55
8  10584  559  10585  10586  10587  560  10588  561  10589  562  10590  563  10591  564  10592  10593  565  10594  566  10595  567  10596  568  10597  569  10598  570  5
71  572  10599  573  10600  574  10601  575  10602  576  10603  577  10604  10605  578  10606  579  580  10607  581  10608  582  583  10609  584  10610  585  10611  586
 587  10612  588  589  10613  590  591  592  10614  593  10615  594  10616  10617  10618  10619  10620  10621  10622  10623  10624  10625  10626  10627  10628  10629  10
```

5. Design two threads to display the student name and CAT1 mark by first thread and student name and CAT2 mark by second thread using **struct data type for a single student** (tutorial: https://www.w3schools.blog/how-to-pass-a-struct-value-to-a-pthread-in-c) in this website the header files have not been included - you include it and refer this web site for passing structure and modify as per the question.

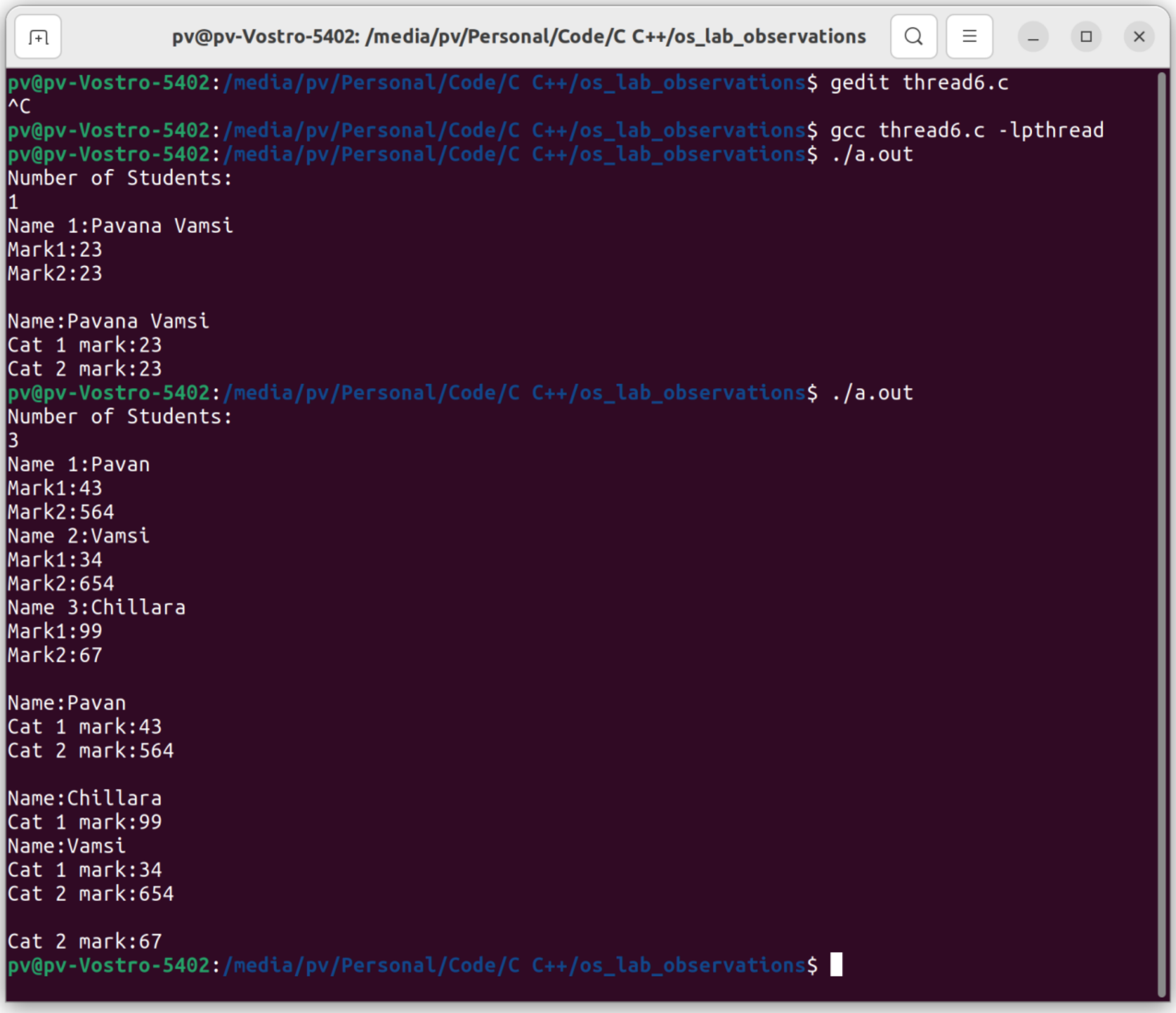

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit thread5.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc thread5.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Name:Pavana Vamsi
Mark1:98
Mark2:99

Name:Pavana Vamsi
Cat 1 mark:98

Name:Pavana Vamsi
Cat 2 mark:99
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```


pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

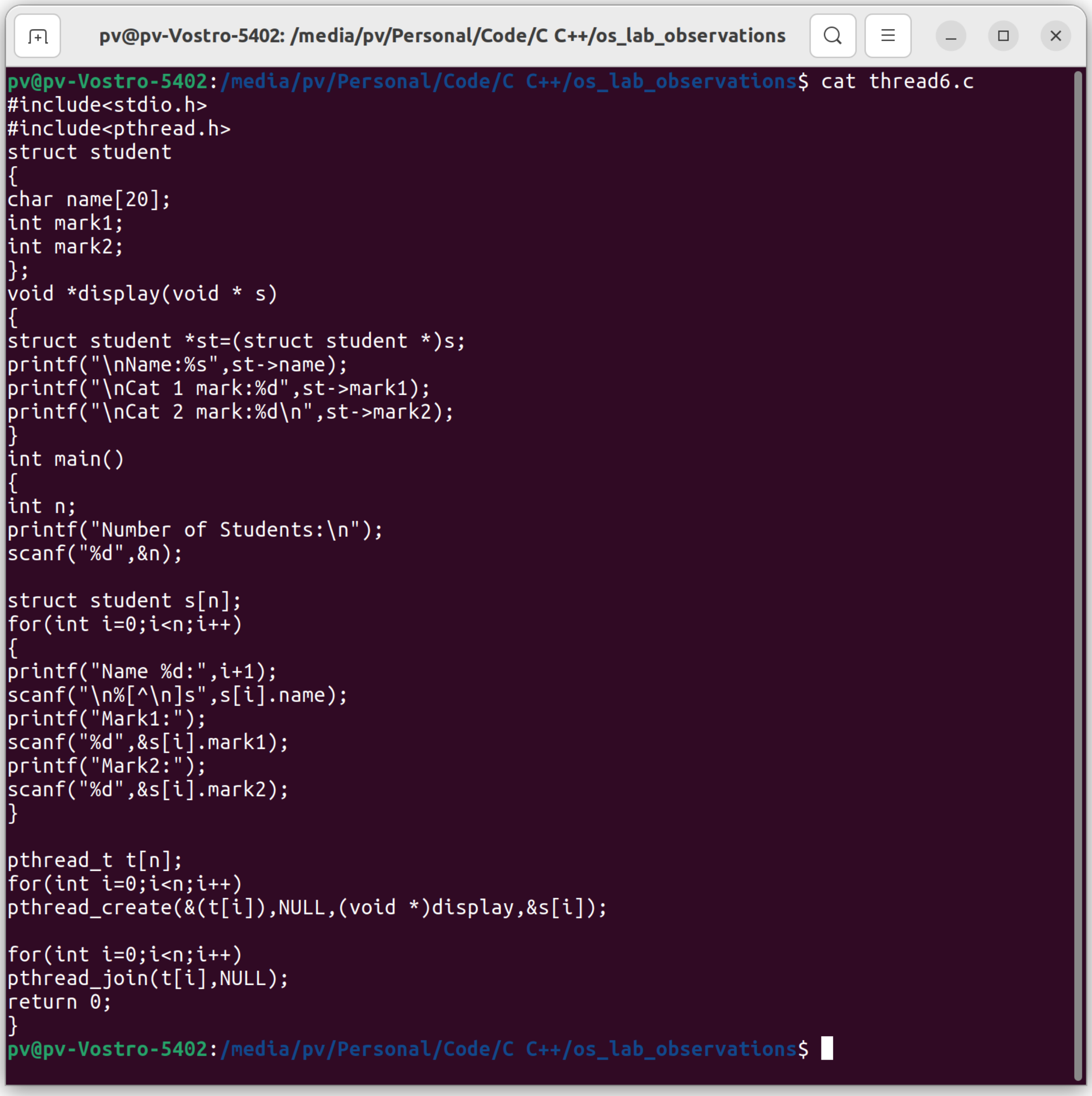
```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat thread5.c
#include<stdio.h>
#include<pthread.h>
struct student
{
char name[20];
int mark1;
int mark2;
};
void *displaycat1(void * s)
{
struct student *st=(struct student *)s;
printf("\nName:%s",st->name);
printf("\nCat 1 mark:%d\n",st->mark1);
}
void *displaycat2(void * s)
{
struct student *st=(struct student *)s;
printf("\nName:%s",st->name);
printf("\nCat 2 mark:%d\n",st->mark2);
}
int main()
{
struct student s1;
printf("Name:");
scanf("%[^\n]s",s1.name);
printf("Mark1:");
scanf("%d",&s1.mark1);
printf("Mark2:");
scanf("%d",&s1.mark2);
pthread_t t1,t2;
pthread_create(&t1,NULL,(void *)displaycat1,&s1);
pthread_create(&t2,NULL,(void *)displaycat2,&s1);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$
```

6. Implement the same program for array of students of class strength 70 and analyse the output generated by the program.



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

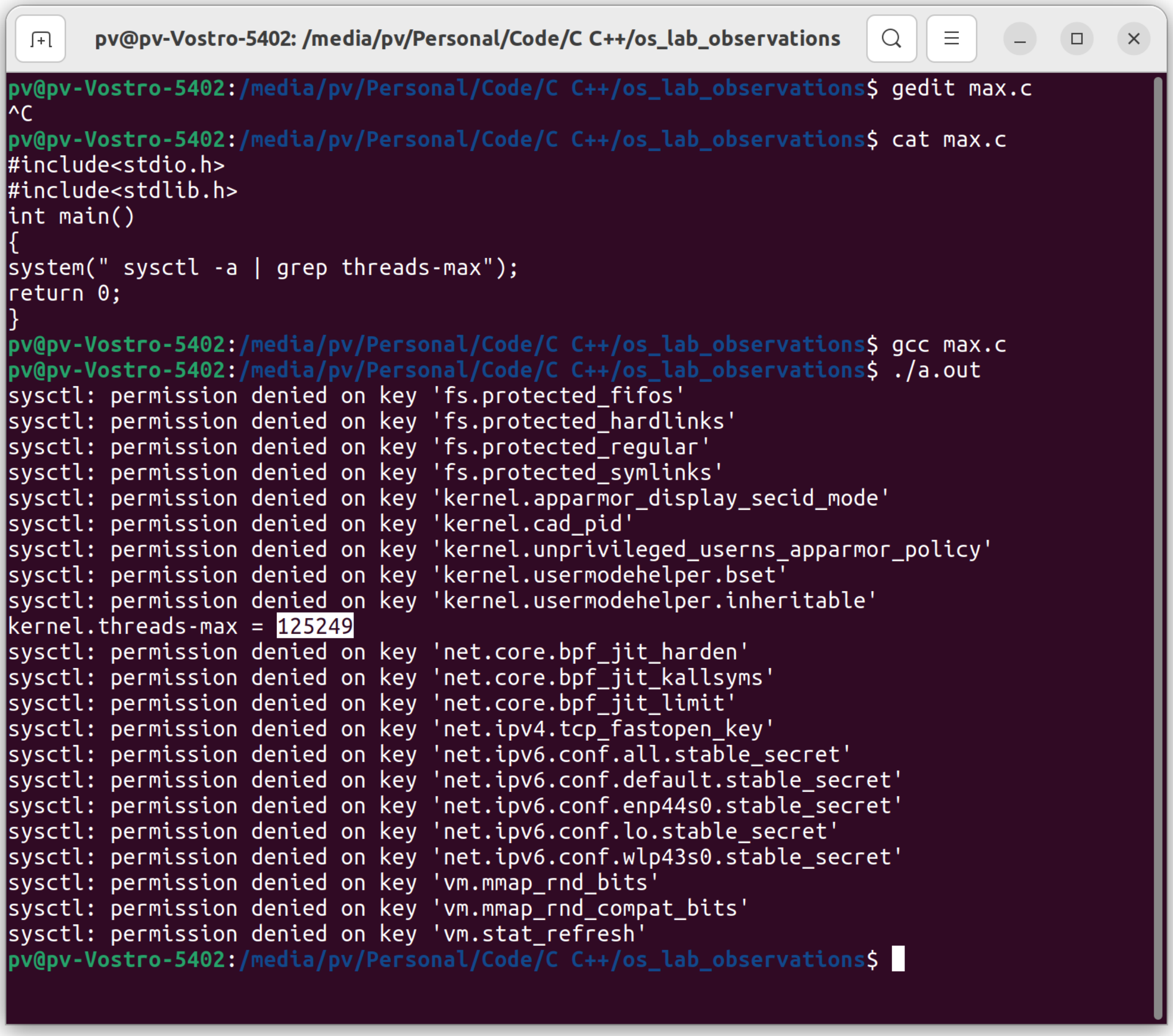
```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit thread6.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc thread6.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Number of Students:
1
Name 1:Pavana Vamsi
Mark1:23
Mark2:23

Name:Pavana Vamsi
Cat 1 mark:23
Cat 2 mark:23
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Number of Students:
3
Name 1:Pavan
Mark1:43
Mark2:564
Name 2:Vamsi
Mark1:34
Mark2:654
Name 3:Chillara
Mark1:99
Mark2:67

Name:Pavan
Cat 1 mark:43
Cat 2 mark:564

Name:Chillara
Cat 1 mark:99
Name:Vamsi
Cat 1 mark:34
Cat 2 mark:654

Cat 2 mark:67
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat thread6.c
#include<stdio.h>
#include<pthread.h>
struct student
{
char name[20];
int mark1;
int mark2;
};
void *display(void * s)
{
struct student *st=(struct student *)s;
printf("\nName:%s",st->name);
printf("\nCat 1 mark:%d",st->mark1);
printf("\nCat 2 mark:%d\n",st->mark2);
}
int main()
{
int n;
printf("Number of Students:\n");
scanf("%d",&n);

struct student s[n];
for(int i=0;i<n;i++)
{
printf("Name %d:",i+1);
scanf("\n%[^\n]s",s[i].name);
printf("Mark1:");
scanf("%d",&s[i].mark1);
printf("Mark2:");
scanf("%d",&s[i].mark2);
}

pthread_t t[n];
for(int i=0;i<n;i++)
pthread_create(&(t[i]),NULL,(void *)display,&s[i]);

for(int i=0;i<n;i++)
pthread_join(t[i],NULL);
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```

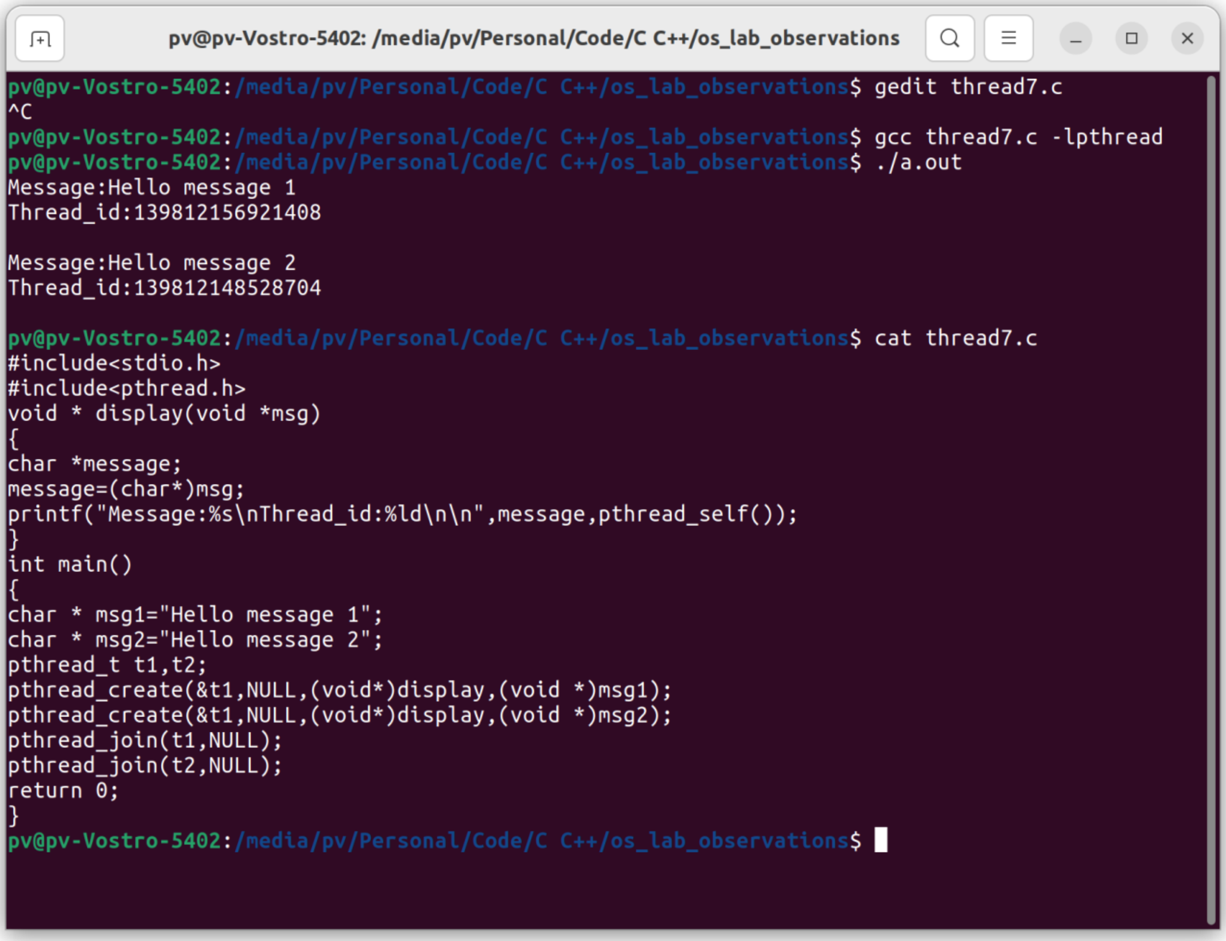
7. Find out the maximum number of threads your system can generate?



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit max.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat max.c
#include<stdio.h>
#include<stdlib.h>
int main()
{
system(" sysctl -a | grep threads-max");
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc max.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
sysctl: permission denied on key 'fs.protected_fifos'
sysctl: permission denied on key 'fs.protected_hardlinks'
sysctl: permission denied on key 'fs.protected_regular'
sysctl: permission denied on key 'fs.protected_symlinks'
sysctl: permission denied on key 'kernel.apparmor_display_secid_mode'
sysctl: permission denied on key 'kernel.cad_pid'
sysctl: permission denied on key 'kernel.unprivileged_userns_apparmor_policy'
sysctl: permission denied on key 'kernel.usermodehelper.bset'
sysctl: permission denied on key 'kernel.usermodehelper.inheritable'
kernel.threads-max = 125249
sysctl: permission denied on key 'net.core.bpf_jit_harden'
sysctl: permission denied on key 'net.core.bpf_jit_kallsyms'
sysctl: permission denied on key 'net.core.bpf_jit_limit'
sysctl: permission denied on key 'net.ipv4.tcp_fastopen_key'
sysctl: permission denied on key 'net.ipv6.conf.all.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.default.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.enp44s0.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.lo.stable_secret'
sysctl: permission denied on key 'net.ipv6.conf.wlp43s0.stable_secret'
sysctl: permission denied on key 'vm.mmap_rnd_bits'
sysctl: permission denied on key 'vm.mmap_rnd_compat_bits'
sysctl: permission denied on key 'vm.stat_refresh'
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ 
```

8. Create two threads and display the two messages along with the corresponding thread_id. (pthread_self () function returns thread id)



pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations

```
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit thread7.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc thread7.c -lpthread
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Message:Hello message 1
Thread_id:139812156921408

Message:Hello message 2
Thread_id:139812148528704

pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat thread7.c
#include<stdio.h>
#include<pthread.h>
void * display(void *msg)
{
char *message;
message=(char*)msg;
printf("Message:%s\nThread_id:%ld\n\n",message,pthread_self());
}
int main()
{
char * msg1="Hello message 1";
char * msg2="Hello message 2";
pthread_t t1,t2;
pthread_create(&t1,NULL,(void*)display,(void *)msg1);
pthread_create(&t1,NULL,(void*)display,(void *)msg2);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$
```