

1. Write a Hello World program using Pthreads.

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

```
void *PrintHello(void *threadid)
{
    long tid;
    tid = (long)threadid;
    printf("Hello World! It's me, thread #%ld!\n", tid);
    pthread_exit(NULL);
}
```

```
int main(int argc, char *argv[])
{
    int NUM_THREADS = atoi(argv[1]);
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;
    for(t=0;t<NUM_THREADS;t++){
        rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
    }
```

```
/* Last thing that main() should do */
pthread_exit(NULL);
}
```

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'labwork5' with files 'a.c' and 'b.c'. The main editor displays the content of 'a.c', which is a C program using pthreads. The code includes headers for pthread.h, stdio.h, and stdlib.h. It defines a function 'PrintHello' that takes a thread ID and prints a message. The main function calls 'PrintHello' in a loop. The terminal at the bottom shows the command 'a 7' being executed, resulting in seven lines of output: 'Hello World! It's me, thread #0!' through 'Hello World! It's me, thread #6!'.

```
a.c - labwork5 - Visual Studio Code
1  #include <pthread.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5
6  void *PrintHello(void *threadid)
7  {
8      long tid;
9      tid = (long)threadid;
10     printf("Hello World! It's me, thread %ld!\n", tid);
11     pthread_exit(NULL);
12 }
13
14 int main(int argc, char *argv[])
15 {
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
dmitriy@dmitriy-Extensa-2510:~/code/parralel/labwork5$ ./a 7
Hello World! It's me, thread #0!
Hello World! It's me, thread #1!
Hello World! It's me, thread #2!
Hello World! It's me, thread #3!
Hello World! It's me, thread #4!
Hello World! It's me, thread #5!
Hello World! It's me, thread #6!
dmitriy@dmitriy-Extensa-2510:~/code/parralel/labwork5$
```

2. Implement a matrix-vector multiplication using Pthreads.

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

```
#define M 3
#define K 2
#define N 3
```

```
int A [M][K];
int B [K][N];
int C [M][N];
```

```
struct v {
int i;
int j;
};
```

```

void *runner(void *param);

int main(int argc, char *argv[])
{

for(int i = 0; i < M;i++)
{
printf("Enter %d row\n", i+1);
for(int j = 0; j < K;j++)
{
scanf("%d",&A[i][j]);
// B[j] [i] =rand()%20;
}
}

for(int i = 0; i < K;i++)
{
printf("Enter %d row\n", i+1);
for(int j = 0; j < N;j++)
{
scanf("%d",&B[i][j]);
// B[j] [i] =rand()%20;
}
}

int i,j, count = 0;
for(i = 0; i < M; i++) {
for(j = 0; j < N; j++) {

struct v *data = (struct v *) malloc(sizeof(struct v));
data->i = i;
data->j = j;

pthread_t tid;
pthread_attr_t attr;

pthread_attr_init(&attr);

pthread_create(&tid,&attr,runner,data);

pthread_join(tid, NULL);
count++;
}
}

```

```

for(i = 0; i < M; i++) {
for(j = 0; j < N; j++) {
printf("%d ", C[i][j]);
}
printf("\n");
}
}

```

```

void *runner(void *param) {
struct v *data = param;
int n, sum = 0;

```

```

for(n = 0; n < K; n++){
sum += A[data->i][n] * B[n][data->j];
}
C[data->i][data->j] = sum;
pthread_exit(0);
}

```

The screenshot shows the Visual Studio Code interface with a C file named `b.c` open in the editor. The code in the editor is as follows:

```

29
30
31     scanf("%d", &A[i][j]);
32     // B[j] [i] = rand()%20;
33 }
34
35 for(int i = 0; i < K; i++)
36 {
37     printf("Enter %d row\n", i+1);
38
39     for(int j = 0; j < N; j++)
40     {
41         scanf("%d", &B[i][j]);
42         // B[j] [i] = rand()%20;

```

The Explorer sidebar on the left shows the project structure with files like `1.png`, `2.png`, `a`, `a.c`, `b`, `b.c`, and PDF files. The TERMINAL panel at the bottom shows the execution of the program with the following output:

```

dmitriy@dmitriy-Extensa-2510:~/code/parralel/labwork5$ ./b
Enter 1 raw
1 2
Enter 2 raw
3 4
Enter 3 raw
5 6
Enter 1 raw
1 2 3
Enter 2 raw
4 5 6
9 12 15
19 26 33
29 40 51
dmitriy@dmitriy-Extensa-2510:~/code/parralel/labwork5$

```

The status bar at the bottom indicates the current file is `main(int argc, char * argv[])` at line 39, column 29, with 4 spaces, UTF-8 encoding, LF line endings, and the C language on a Linux system.