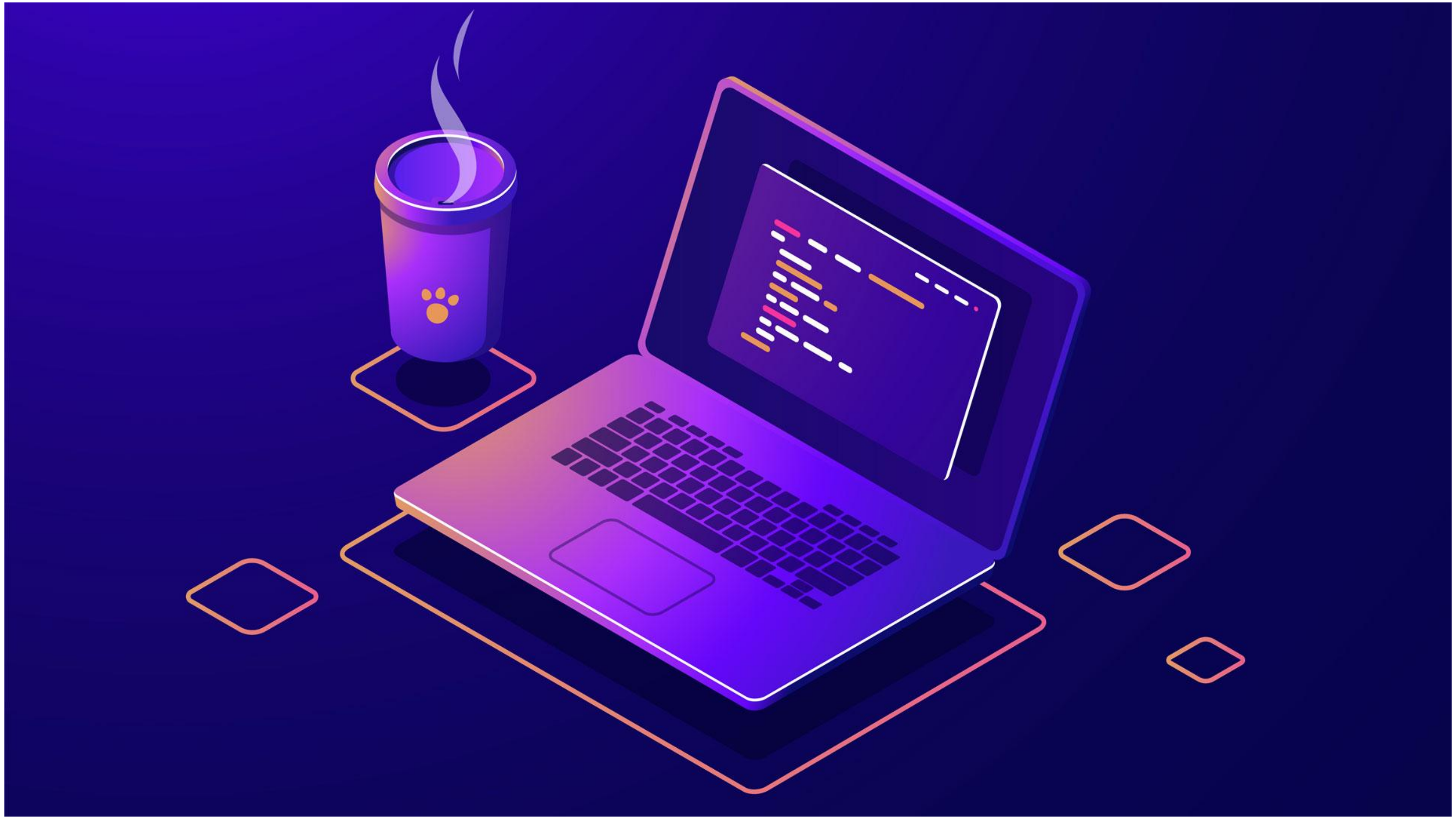
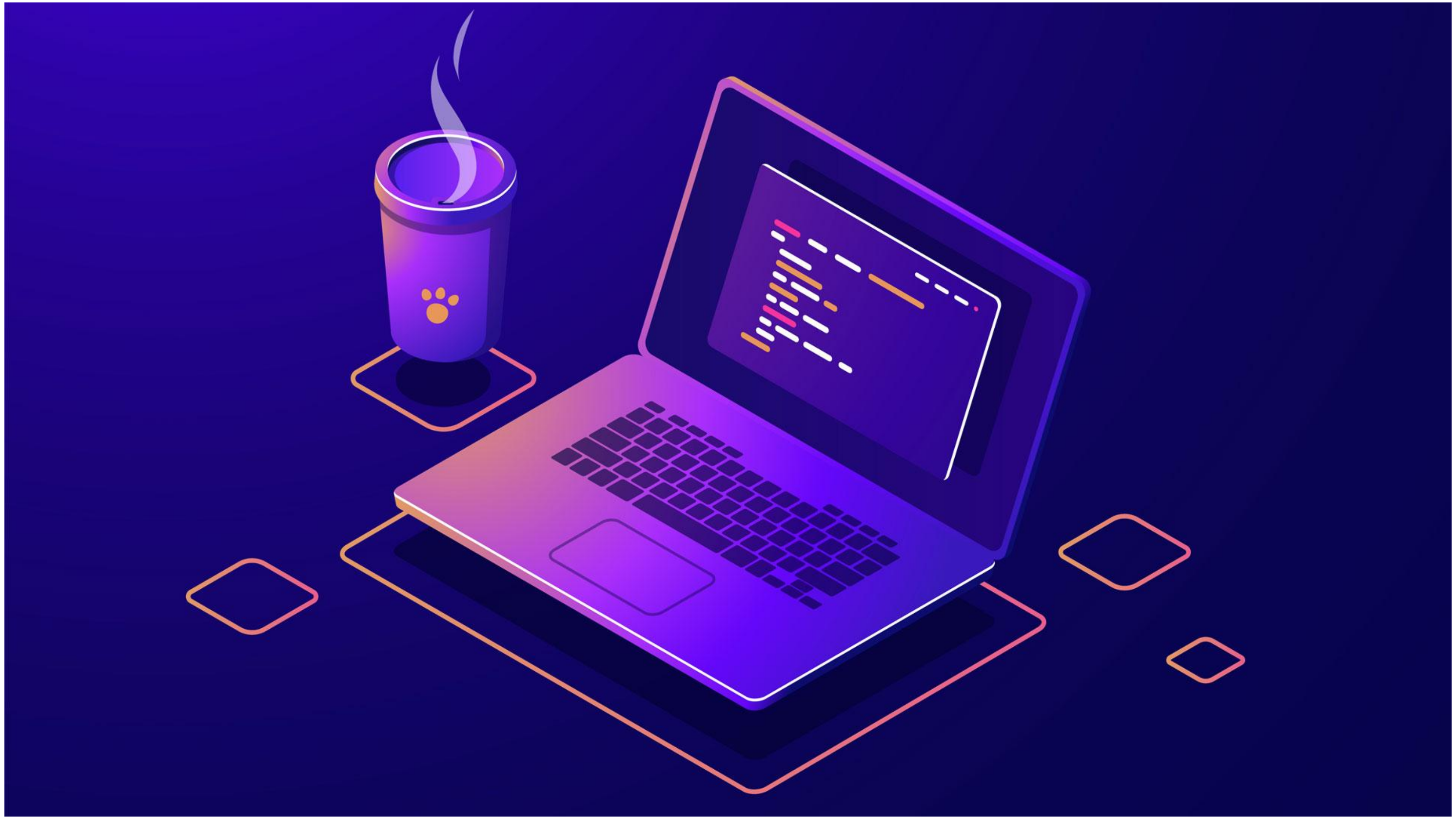
OpenMP Problems



Problem 4



Problem 4

SAXPY (*Single-precision A times X Plus Y*) is a combination of scalar multiplication and vector addition. It takes as input two vectors of 32-bit floats x and y with n elements each, and a scalar value a. It multiplies each element $x[i]$ by a and adds the result to $y[i]$, as shown below:

```
void saxpy(int n, float a, float *x, float *y) {
        for (int i = 0; i < n; ++i) y[i] = a * x[i] + y[i];
}
```

Problem 4

Given the following main program that makes use of saxpy function already defined. We want to achieve the asynchronous execution of the initialization loop, the two SAXPY calls and the two loops writing the result vectors to files.

Write a parallel version making use of the OpenMP task construct and task dependencies, creating the appropriate parallel context.

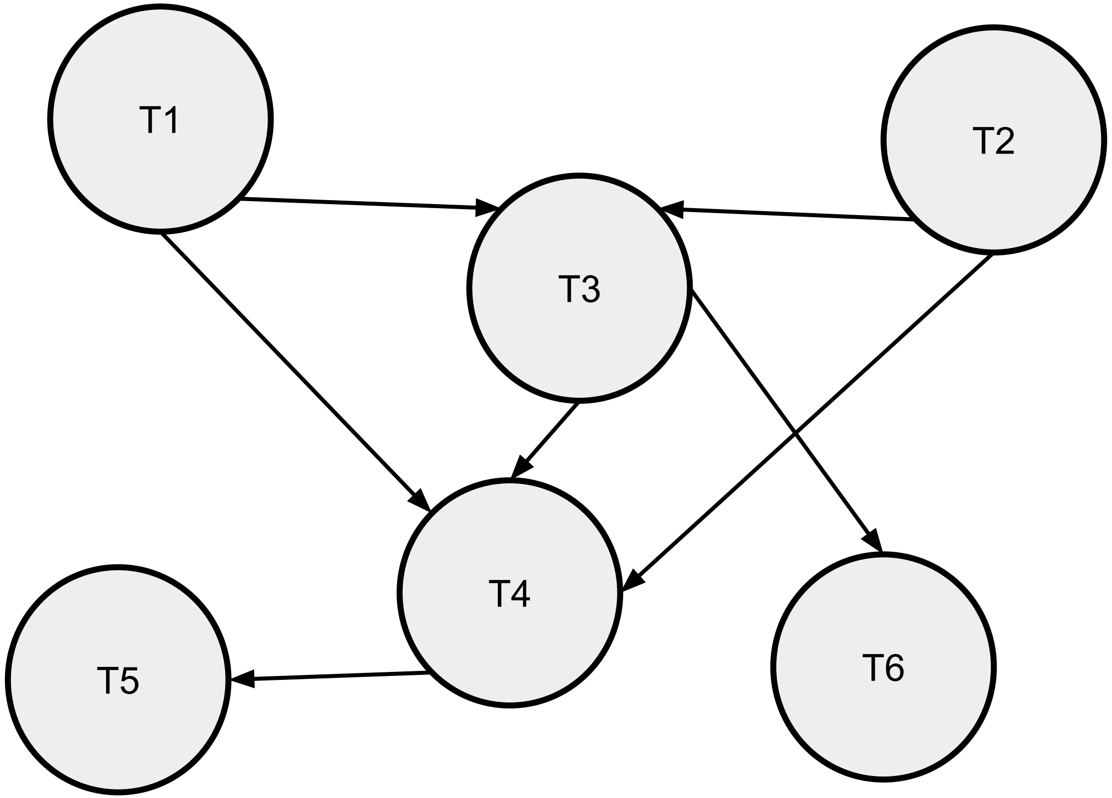
Problem 4

```
int main() {
        int N = 1 << 20;    /* 1 million floats */
        float  *fx = (float *) malloc(N * sizeof(float));
        float  *fy = (float *) malloc(N * sizeof(float));
        FILE *fpx = fopen("fx.out", "w");
        FILE *fpy = fopen("fy.out", "w");
        /* simple initialization just for testing */
        for (int k = 0; k < N; ++k)
                fx[k] = 2.0f + (float) k;
        for (int k = 0; k < N; ++k)
                fy[k] = 1.0f + (float) k;
        /* Run SAXPY TWICE */
        saxpy(N, 3.0f, fx, fy);
        saxpy(N, 5.0f, fy, fx);

        /* Save results */
        for (int k = 0; k < N; ++k)
                fprintf(fpx, " %f ", fx[k]);
        for (int k = 0; k < N; ++k)
                fprintf(fpy, " %f ", fy[k]);

        free(fx); fclose(fpx);
        free(fy); fclose(fpy);
}
```

Problem 4



Lucas Bazilio - Udemy

Instructor Social Media

Youtube: Lucas Science

Instagram: lucaasbazilio

Twitter: lucasebazilio