Report on task 1 (template)

**Name: Sarsenov Damir**

**Group: BDA-2008**

**E-mail: immbishe@gmail.com**

**Main part:**

# Step 1: clone Mini-Rt repository

Repository: <https://github.com/georgy-schukin/mini-rt>

Command:

git clone https://github.com/georgy-schukin/mini-rt

A screenshot of a computer

Description automatically generated with medium confidence

# Step 2: build and install Mini-Rt library

If you are working on our public HPC server: nothing to do, the library is already installed!

If you are working on your own Linux machine:

First, make sure cmake and make programs are installed. Then go to the directory where Mini-Rt repository was cloned and execute next commands:

mkdir build

cd build

cmake ../mini-rt/src

make

sudo make install

By default the library will be installed in /usr/local directory (include files in /usr/local/include/minirt directory, lib files - in /usr/local/lib directory).

If you are working on Windows machine:

Install cmake for Windows and build the library with it (sources are located in src folder).

# Step 3: build sequential ray tracing application

Create a directory for your project. Copy example application minirt\_test.cpp from mini-rt/src/test directory to your project directory, rename it to raytracing.cpp:

cd *<your project dir>*

cp *<mini-rt dir>*/src/test/minirt\_test.cpp raytracing.cpp

Build the application:

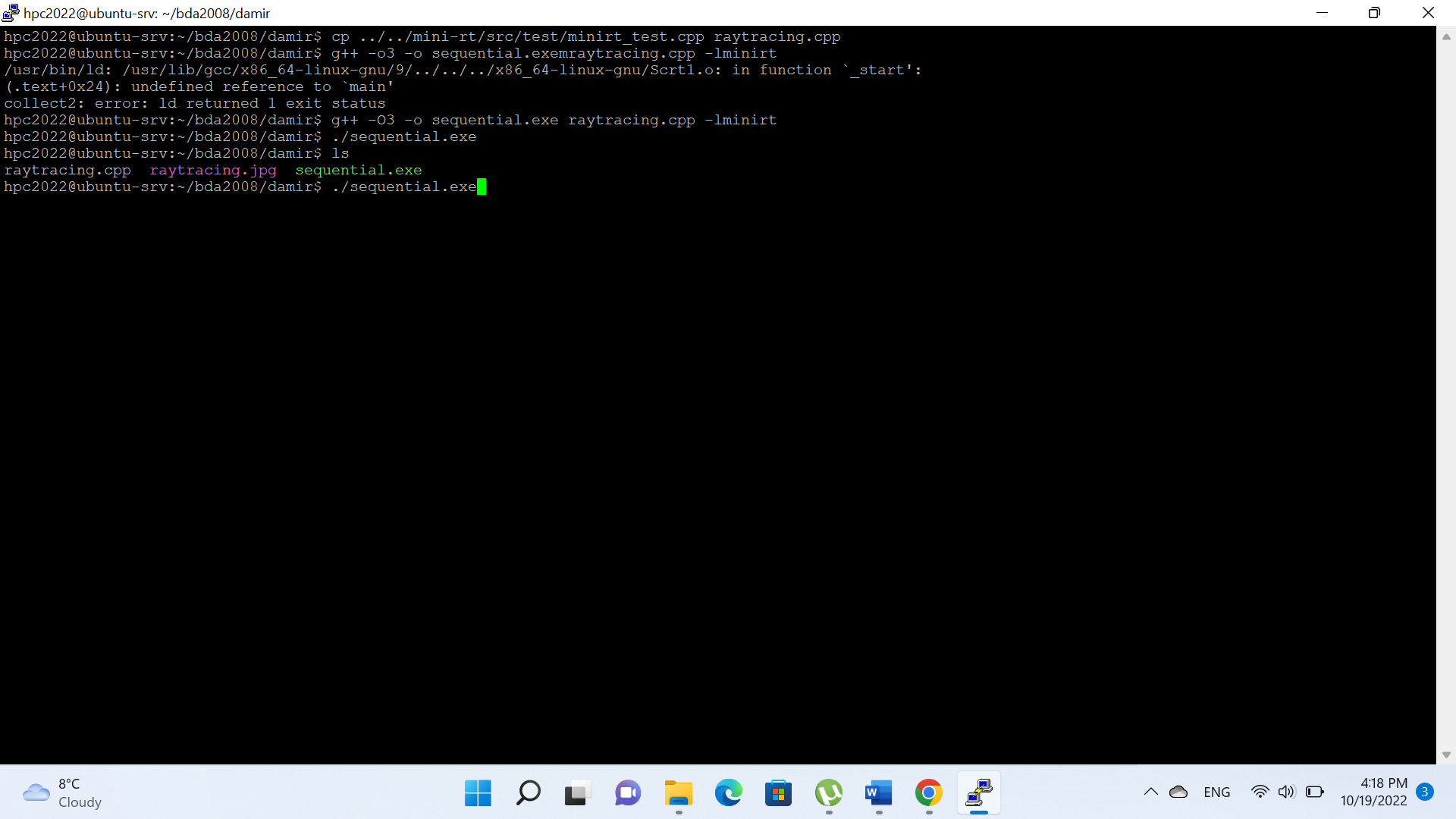
g++ -O3 -o sequential.exe raytracing.cpp -lminirt

Run the application:

./sequential.exe

If everything is done correctly, the resulting .jpg image file will be created.

<make screenshots>



# Step 4: play with the application

The default test application accepts parameters via command line (if some parameters are not given, default values are used). Parameters are:

./sequential.exe <Image resolution by X> <Image resolution by Y> <Number of samples>

Change such parameters as image resolution or number of samples per pixel (command line arguments 1, 2 and 3 by default), observe the effect on the resulting image:

./sequential.exe 640 480 1

./sequential.exe 1024 768 1

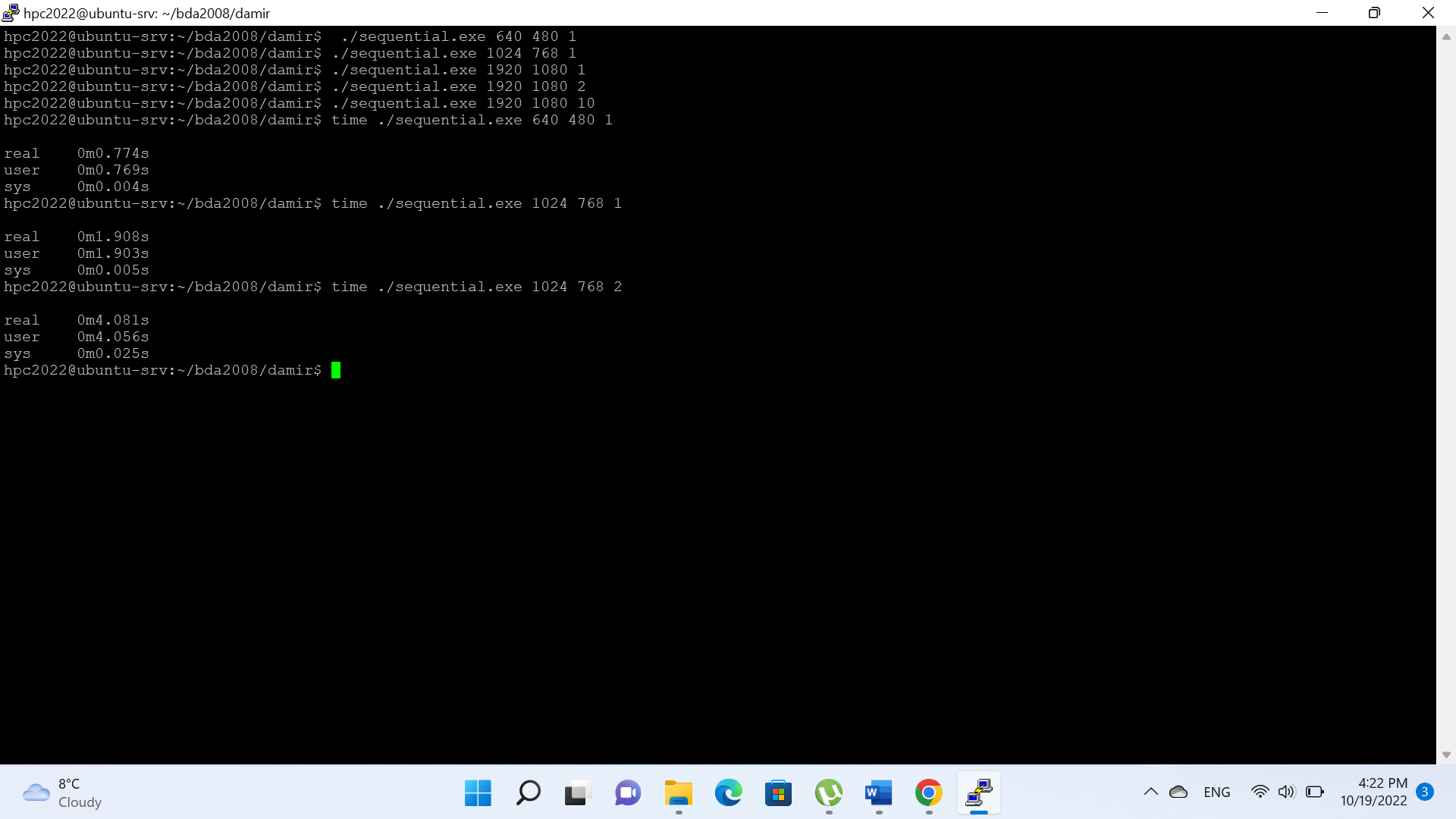
./sequential.exe 1920 1080 1

./sequential.exe 1920 1080 2

./sequential.exe 1920 1080 10

Measure running time of the application with different arguments (image resolution, number of samples, etc). You can use time command to measure time:

time ./sequential.exe *<arguments>*



# Step 5: create git repository on Github for this course, upload your project files for Task 1 to it, include a link to the repository in the report

Graphical user interface, text, application, chat or text message

Description automatically generated

<https://github.com/Flink-alt/HPC>