Setting Sycl Enviroment

This guide assumes the usage of Ubuntu 22.04 LTS.

1. Install Docker

- Credits: digitalocean.com

Step 1: Update Your Existing List of Packages

```
sudo apt update
```

Step 2: Install a Few Prerequisite Packages which Let apt Use Packages over HTTPS

```
sudo apt install apt-transport-https ca-certificates curl
↪  software-properties-common
```

Step 3: Add the GPG Key for the Official Docker Repository to Your System

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
↪  --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Step 4: Add the Docker Repository to APT Sources

```
echo "deb [arch=amd64
↪  signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
↪  https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" |
↪  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Step 5: Update Your Existing List of Packages Again

```
sudo apt update
```

Step 6: Make Sure You are About to Install from the Docker Repository Instead of the Default Ubuntu Repository

```
apt-cache policy docker-ce
```

Step 7: Install Docker

```
sudo apt install docker-ce
```

Step 8: Check Docker Status

```
sudo systemctl status docker
```

2. Pulling the `intel/oneapi-basekit` Image

Pull the Image from Docker Hub

```
docker pull intel/oneapi-basekit
```

3. Running the `intel/oneapi-basekit` Image

In order to test that the SYCL environment is working properly, let's try compiling and running the hello-world code presented in this lesson. The code is available on the workshop webpage.

Download and extract the code and then follow the following steps:

```
cd CODE_DIRECTORY
image=intel/oneapi-basekit
docker run --device=/dev/dri --privileged -it -v "$(pwd)":/app:Z "$image"
```

4. Compile Sycl Code

In order to compile the code you simply `cd into app/` and then you can compile it using:

```
icpx -fsycl hello-world.cpp -o hello-world
```

or you can compile the file using cmake:

```
mkdir build
cd build
cmake ..
make
```

And to run:

```
./hello-world
```

5. Managing Docker Containers

List Running Containers

```
docker ps
```

List All Containers (Including Stopped Containers)

```
docker ps -a
```

Stop a Running Container

```
docker stop <container_id>
```

Remove a Container

```
docker rm <container_id>
```

Remove an Image

```
docker rmi <image_id>
```

Conclusion

You have now installed Docker, pulled the `intel/oneapi-basekit` image, and launcher! This is all for this first class. In the next one, we will start using SYCL.

Bellow there are a couple of exercises for you to remember cpp.

CPP Exercises

1. Write an hello world

```cpp
#include<iostream>

int main(){
  // Print hello world
  return 0;
}
```

2. Write a lambda function that squares a given number

```cpp
int main(){
  // Fill in the blanks here
  auto square = [](  )
     {

      // return something

     };

   int a = square(2);

   return 0;
}
```

3. Lambda function that returns Sum of all elements of the given vector

```cpp
#include<iostream>

int main(){
   std::vector<int> a {1,2,3,4,5,6,7,9,11,16,19};
   int s=0;

  // Fill in the blanks here
  std::for_each(a.begin(), a.end(), [   ] (   ) {      } );


  std::cout << s << std::endl;
  return 0;
}
```