

```
1 #include <stdio.h>
2
3 // OpenCL Headers
4 #include <CL/openc1.h>
5
6 int main(void)
7 {
8     // function declarations
9     void printOpenCLDeviceProperties(void);
10
11     // code
12     printOpenCLDeviceProperties();
13 }
14
15 void printOpenCLDeviceProperties(void)
16 {
17     // code
18     printf("OpenCL INFORMATION :\n");
19     printf("=====
20     =====\n");
21
22     cl_int result;
23     cl_platform_id ocl_platform_id;
24     cl_uint dev_count;
25     cl_device_id *ocl_device_ids;
26     char oclPlatformInfo[512];
27
28     // get first platform ID
29     result=clGetPlatformIDs(1,&ocl_platform_id,NULL);
30     if(result != CL_SUCCESS)
31     {
32         printf("clGetPlatformIDs() Failed\n");
33         exit(EXIT_FAILURE);
34     }
35
36     // get GPU device count
37     result=clGetDeviceIDs
38     (ocl_platform_id,CL_DEVICE_TYPE_GPU,0,NULL,&dev_count);
39     if(result != CL_SUCCESS)
40     {
41         printf("clGetDeviceIDs() Failed\n");
42         exit(EXIT_FAILURE);
43     }
44     else if(dev_count==0)
45     {
46         printf("There Is No OpenCL Supprted Device On This System.\n");
47         exit(EXIT_FAILURE);
48     }
49     else
50     {
51         // get platform name
52         clGetPlatformInfo
53         (ocl_platform_id,CL_PLATFORM_NAME,500,&oclPlatformInfo,NULL);
54         printf("OpenCL Supporting GPU Platform Name : %s\n",oclPlatformInfo);
55         // get platform version
```

```

53  clGetPlatformInfo
    (ocl_platform_id,CL_PLATFORM_VERSION,500,&oclPlatformInfo,NULL);
54  printf("OpenCL Supporting GPU Platform Version : %s
    \n",oclPlatformInfo);
55  // print supporting device number
56  printf("Total Number Of OpenCL Supporting GPU Device/Devices On This
    System : %d\n",dev_count);
57  // allocate memory to hold those device ids
58  ocl_device_ids=(cl_device_id *)malloc(sizeof(cl_device_id)*dev_count);
59  // get ids into allocated buffer
60  clGetDeviceIDs
    (ocl_platform_id,CL_DEVICE_TYPE_GPU,dev_count,ocl_device_ids,NULL);
61  char ocl_dev_prop[1024];
62  int i;
63  for(i=0;i<(int)dev_count;i++)
64  {
65      printf("\n");
66      printf("***** GPU DEVICE GENERAL INFORMATION *****\n");
67      printf("===== \n");
68      printf("GPU Device Number                : %d\n",i);
69      clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_NAME,sizeof
    (ocl_dev_prop),&ocl_dev_prop,NULL);
70      printf("GPU Device Name                : %s
    \n",ocl_dev_prop);
71      clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_VENDOR,sizeof
    (ocl_dev_prop),&ocl_dev_prop,NULL);
72      printf("GPU Device Vendor                : %s
    \n",ocl_dev_prop);
73      clGetDeviceInfo(ocl_device_ids[i],CL_DRIVER_VERSION,sizeof
    (ocl_dev_prop),&ocl_dev_prop,NULL);
74      printf("GPU Device Driver Version        : %s
    \n",ocl_dev_prop);
75      clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_VERSION,sizeof
    (ocl_dev_prop),&ocl_dev_prop,NULL);
76      printf("GPU Device OpenCL Version        : %s
    \n",ocl_dev_prop);
77      cl_uint clock_frequency;
78      clGetDeviceInfo(ocl_device_ids
    [i],CL_DEVICE_MAX_CLOCK_FREQUENCY,sizeof
    (clock_frequency),&clock_frequency,NULL);
79      printf("GPU Device Clock Rate            : %u
    \n",clock_frequency);
80
81      printf("\n");
82      printf("***** GPU DEVICE MEMORY INFORMATION *****\n");
83      printf("===== \n");
84      cl_ulong mem_size;
85      clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_GLOBAL_MEM_SIZE,sizeof
    (mem_size),&mem_size,NULL);
86      printf("GPU Device Global Memory          : %llu
    Bytes\n",(unsigned long long)mem_size);
87      cl_device_local_mem_type local_mem_type;
88      clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_LOCAL_MEM_SIZE,sizeof
    (mem_size),&mem_size,NULL);
89      printf("GPU Device Local Memory          : %llu
    Bytes\n",(unsigned long long)mem_size);

```

```

\HPP Seminar 2022\openc1\DevProp.c 3
100 clGetDeviceInfo(ocl_device_ids[i], 7
    CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE,sizeof 7
    (mem_size),&mem_size,NULL);
101 printf("GPU Device Constant Buffer Size : %llu 7
    Bytes\n",(unsigned long long)mem_size);
102
103 printf("\n");
104 printf("***** GPU DEVICE COMPUTE INFORMATION 7
    *****\n");
105 printf 7
    ("===== 7
    \n");
106 cl_uint compute_units;
107 clGetDeviceInfo(ocl_device_ids 7
    [i],CL_DEVICE_MAX_COMPUTE_UNITS,sizeof 7
    (compute_units),&compute_units,NULL);
108 printf("GPU Device Number Of Parallel Processors Cores : %u 7
    \n",compute_units);
109 size_t workgroup_size;
110 clGetDeviceInfo(ocl_device_ids 7
    [i],CL_DEVICE_MAX_WORK_GROUP_SIZE,sizeof 7
    (workgroup_size),&workgroup_size,NULL);
111 printf("GPU Device Work Group Size : %u\n", 7
    (unsigned int)workgroup_size);
112 size_t workitem_dims;
113 clGetDeviceInfo(ocl_device_ids 7
    [i],CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS,sizeof 7
    (workitem_dims),&workitem_dims,NULL);
114 printf("GPU Device Work Item Dimensions : %u\n", 7
    (unsigned int)workitem_dims);
115 size_t workitem_size[3];
116 clGetDeviceInfo(ocl_device_ids 7
    [i],CL_DEVICE_MAX_WORK_ITEM_SIZES,sizeof 7
    (workitem_size),&workitem_size,NULL);
117 printf("GPU Device Work Item Sizes : %u/%u/%u 7
    \n",(unsigned int)workitem_size[0],(unsigned int)workitem_size 7
    [1],(unsigned int)workitem_size[2]);
118 }
119 free(ocl_device_ids);
120 }
121 }
122

```