OpenCL Probe

Logan Streondj

September 26, 2016
Version 0.1

Abstract

Demonstrates OpenCL platform device knowledge xor tries to compile a .cl file based on input filename.

1 Synopsis

clprobe input filename

2 Description

if there is no filename ingredient then *clprobe* demonstrates openCL which includes the platform and instrument knowledge.

```
if (ingredient_long != 2) {
  fprintf(stderr,
          "no .cl filename ingredient so demonstrating openCL knowledge\n");
  getInfo();
  exit(0);
}
```

Otherwise Progclprobe accepts one ingredient, which is the input filename.

```
  char *filename = ingredient_list[1];
```

It then sets up a minimal OpenCL environment, and attempts to compile it.

```
  cl_platform_id platform_id = NULL;
  cl_uint ret_num_platforms;
  cl_context context = 0;
  cl_program program = 0;
  cl_device_id device_id = 0;
  cl_uint ret_num_devices;
  cl_int return_number;

  return_number = clGetPlatformIDs(1, &platform_id, &ret_num_platforms);
  if (!success_verification(return_number)) {
    fprintf(stderr, "Failed to get platform id's. %s:%d\n", __FILE__, __LINE__);
    return 1;
  }
  return_number = clGetDeviceIDs(platform_id, CL_DEVICE_TYPE_DEFAULT, 1,
                                 &device_id, &ret_num_devices);
  if (!success_verification(return_number)) {
    fprintf(stderr, "Failed to get OpenCL devices. %s:%d\n", __FILE__,
            __LINE__);
    return 1;
  }
  context = clCreateContext(NULL, 1, &device_id, NULL, NULL, &return_number);
  if (!success_verification(return_number)) {
    fprintf(stderr, "Failed to create an OpenCL context. %s:%d\n", __FILE__,
            __LINE__);
    return 1;
  }
  seed_program_probe(device_id, context, filename, &program);
```

3 Bugs

Still need to add ability to load header files.

4 Author

Logan Streondj ¡streondj at gmail dot com¿

5 See Also

generic.c *libOpenCL*(7)

6 License