

Tartalomjegyzék

Tartalomjegyzék	3
1. Bevezetés	8
1.1. Az OpenCL története	8
1.2. Az OpenCL jelene és jövője	14
1.3. OpenCL a Flynn-osztályokban	14
1.4. Mire használható az OpenCL?	16
1.5. A jegyzetről	18
1.5.1. A jegyzet célja és használata	18
1.5.2. A jegyzet alapja	19
1.5.3. Hardverkörnyezet	20
2. Első OpenCL-programunk	21
2.1. Programozási környezet	21
2.2. Hello world	22
2.3. Fordítás, linkelés, futtatás	27
2.4. Összefoglalás	28
3. Az OpenCL-modell	29
3.1. Az OpenCL-specifikáció	29
3.1.1. Platformmodell	30
3.1.2. Végrehajtási modell	30
3.1.3. Memóriamodell	34
3.1.4. Programozási modell	39
3.2. Összefoglalás	40
3.3. Feladatok	40
4. Az OpenCL API	41
4.1. Hibakezelés	41
4.2. Retain/Release	43
4.3. Elnevezési konvenciók	44
4.4. Platformréteg	44
4.4.1. Platformok azonosítása	45
4.4.2. Eszközök, aleszközök	48
4.4.3. Környezet	57
4.5. Futtatóréteg	62
4.5.1. Parancssor	63

4.5.2.	Eseményobjektumok	63
4.5.3.	Pufferobjektumok és a memóriakezelés	66
4.5.4.	Szinkronizáció	80
4.5.5.	Programobjektumok	85
4.5.6.	Kernelobjektumok	104
4.5.7.	Kernelek végrehajtása	109
4.6.	Összefoglalás	118
4.7.	Feladatok	118
5.	Az OpenCL C nyelv	120
5.1.	Az OpenCL C programozási nyelv	120
5.2.	Típusok	120
5.2.1.	Beépített atomi típusok	120
5.2.2.	Beépített vektortípusok	121
5.2.3.	Konverzió	123
5.3.	Kifejezések	125
5.3.1.	Aritmetikai konverzió	125
5.3.2.	Operátorok	125
5.4.	Minősítők	127
5.5.	Vezérlési szerkezetek	131
5.6.	Könyvtári függvények	131
5.6.1.	Munkaelemfüggvények	131
5.6.2.	Matematikai függvények	133
5.6.3.	Geometriai függvények	136
5.6.4.	Relációs függvények	137
5.6.5.	Vektorok betöltéséhez és kiírásához kapcsolódó függvények	137
5.6.6.	Szinkronizációs függvények	138
5.6.7.	Aszinkron memóriakezelő függvények	140
5.6.8.	Atomi függvények	142
5.6.9.	<code>printf</code>	144
5.7.	Összefoglalás	144
5.8.	Feladatok	144
6.	Esettanulmányok: Lineáris algebra, mátrixszorzás	146
6.1.	Matematikai háttér	146
6.2.	Időmérés	149
6.3.	CPU-implementáció	150
6.4.	Naiv OpenCL-implementáció	152
6.5.	OpenCL C optimalizáció	159
6.6.	A párhuzamosítás finomságának csökkentése	159
6.7.	A privát memória használata	162
6.8.	A lokális memória használata	163
6.9.	Összefoglalás	165
6.10.	Feladatok	166

7. Esettanulmány: Digitális képfeldolgozás, konvolúciós szűrés	170
7.1. Elméleti háttér	170
7.2. CPU-implementáció	172
7.3. Naiv OpenCL-implementáció	177
7.4. OpenCL C optimalizálás	179
7.5. A konstans memória használata	180
7.6. A lokális memória használata	181
7.7. Összefoglalás	187
7.8. Feladatok	187
8. Kép- és mintavételező objektumok	190
8.1. Képjelöltek	192
8.1.1. Létrehozás	192
8.1.2. Írás/olvasás, feltöltés	197
8.1.3. Egyéb függvények	200
8.2. Mintavételező objektumok	201
8.3. OpenCL C	203
8.3.1. Típusok	204
8.3.2. Minősítők	204
8.3.3. Létrehozás	204
8.3.4. Függvények	205
8.4. Példa	209
8.5. OpenCL 1.1	213
8.6. Összefoglalás	218
8.7. Feladatok	218
9. Esettanulmány: Statisztika, hisztogramszámítás	220
9.1. Elméleti háttér	220
9.1.1. A hisztogram fogalma	220
9.1.2. Hisztogramok alkalmazásai	222
9.1.3. Adaptív küszöbölés	223
9.2. A hisztogramszámítás CPU-implementációja	225
9.3. Naiv OpenCL-implementáció	226
9.4. Naiv OpenCL-implementáció 2D indextartománnyal	229
9.5. OpenCL-implementáció képjelöltek használatával	230
9.6. A lokális memória használata	233
9.7. Egy másik „leggyorsabb” implementáció	235
9.8. Összefoglalás	240
9.9. Feladatok	240
10. Esettanulmány: Jelfeldolgozás, diszkrét Fourier-transzformáció	243
10.1. Elméleti háttér	243
10.1.1. Komplex számok	243
10.1.2. A diszkrét Fourier-transzformáció bevezetése	244
10.2. Példa	248
10.3. A diszkrét Fourier-transzformáció alkalmazásai	249
10.4. CPU-implementáció	253

10.5. Naiv OpenCL-implementáció	255
10.6. OpenCL C optimalizálás	258
10.7. A lokális memória használata	259
10.8. Vektortípusú változók használata	261
10.9. Vektortípusok és a lokális memória használata	264
10.10 Gyors Fourier-transzformáció	265
10.11 Gyors Fourier-transzformáció és lokális memória használata	271
10.12 Összefoglalás	278
10.13 Feladatok	278
11. Esettanulmány: Fizika, részecskeszimuláció	282
11.1. Elméleti háttér	282
11.2. CPU-implementáció	286
11.3. Naiv OpenCL-implementáció	290
11.4. Vektortípusok használata	293
11.5. OpenCL C optimalizálás	294
11.6. A konstans memória használata	295
11.7. A lokális memória használata	295
11.8. Adatok pakolása	297
11.9. Összefoglalás	299
11.10 Feladatok	299
12. OpenCL-kiterjesztések	301
12.1. Az OpenCL-kiterjesztések szervezése	301
12.2. Az OpenCL-kiterjesztések használata	303
12.3. Interaktív részecskeszimuláció	305
12.3.1. OpenCL és OpenGL együttes használata interoperáció nélkül	305
12.3.2. OpenCL és OpenGL együttes használata interoperációval	311
12.4. Összefoglalás	316
12.5. Feladatok	316
13. Kapcsolódó technológiák	318
13.1. OpenCL és más párhuzamos programozási technológiák együttes alkalmazása	318
13.1.1. Szálbiztosság	318
13.1.2. Összetett alkalmazások	319
13.2. A C++ interfész	320
13.3. NVidia CUDA	320
13.4. WebCL	324
13.5. Feladatok	327
14. Utószó	328
A. cmake	331
A.1. A cmake-ről általában	331
A.2. Első cmake projektünk	333
A.3. OpenCL-alkalmazásunk fordítása cmake-vel	338
A.4. Feladatok	341

B. R	343
B.1. Egyszerű grafikonok készítése R-ben	343
B.2. Feladatok	347
C. Képek írása/olvasása	348
C.1. Portable GrayMap – PGM	348
C.2. Portable Network Graphics – PNG	351
C.3. Feladatok	356
Irodalomjegyzék	357
Tárgymutató	359