

Wydział Informatyki										
Kierunek studiów	Informatyka							Poziom i forma studiów	drugiego stopnia stacjonarne	
Specjalność / Ścieżka dyplomowania	Inżynieria Oprogramowania							Profil kształcenia	ogólnoakademicki	
Nazwa przedmiotu	Obliczenia z wykorzystaniem akceleratorów graficznych							Kod przedmiotu	INF2AKC	
								Rodzaj przedmiotu	obieralny	
Forma zajęć i liczba godzin	W	Ć	L	P	Ps	T	S	Semestr	2	
	15				30			Punkty ECTS	3	
Przedmioty wprowadzające										
Cele przedmiotu	Zapoznanie studentów z metodami implementacji algorytmów na procesory graficzne GPU w technologii CUDA.									
Treści programowe	Wykład: 1. Wprowadzenie w programowanie GPU. Pierwszy program w CUDA. 2. Organizacja wątków w gridy i bloki. 3. Organizacja pamięci, pamięć współdzielona, funkcja _syncthreads 4. Optymalizacja kodu: organizacja wątków w warpy, unikanie rozbieżności, dostęp skojarzony do pamięci (ang. memory coalescing). 5. Studium przypadku: Histogram i operację atomowe. 6. Studium przypadku: Splot dwuwymiarowy. 7. Studium przypadku: redukcja. 8-9. Narzędzie profilujące i ułatwiające usuwanie błędów 10. Biblioteki CUBLAS CUB i inne. 11. Biblioteka Thrust 12. Programowanie wielu układów GPU. 13. Strumienie i współbieżne uruchamianie wielu jąder. 14. Zrównoleglanie dynamiczne (ang. CUDA dynamic parallelizm). 15. Wykorzystanie klastrów akceleratorów w modelu CUDA+MPI Pracownia: 1. Pierwszy program w CUDA 2. Zaawansowane mnożenie macierzy 3. Histogram/redukcja 4. Wykorzystanie bibliotek CUDA 5. Praca z narzędziami cuda-memcheck i nsight compute 6. Programowanie wielu układów GPU i uruchamianie wielu jąder 7. Dynamiczne zrównoleglenie - algorytm quicksort. 8. Wykorzystanie wielu akceleratorów na klastrze.									
Metody dydaktyczne	wykład problemowy, programowanie z użyciem komputera,									
Forma zaliczenia	Wykład - egzamin pisemny albo test zaliczający Pracownia specjalistyczna - Wprowadzenie +7 czterogodzinnych zadań programistycznych.									
Symbol efektu uczenia się	Zakładane efekty uczenia się								Odniesienie do kierunkowych efektów uczenia się	
EU1	zna i rozumie zasady programowania algorytmów równoległych na procesory graficzne GPU, w szczególności w standardzie CUDA								INF2_W03 INF2_W06	
EU2	zna i rozumie metody optymalnego wykorzystania akceleratorów graficznych w obliczeniach ogólnego przeznaczenia								INF2_W03 INF2_W06	
EU3	posiada umiejętność tworzenia algorytmów równoległych na akceleratory graficzne GPU								INF2_U03 INF2_U04 INF2_U08	
EU4	posiada umiejętność optymalizacji algorytmów równoległych przeznaczonych na akceleratory graficzne GPU oraz potrafi wykorzystywać narzędzia wspomagające ten proces								INF2_U03 INF2_U04 INF2_U08	
Symbol efektu uczenia się	Sposób weryfikacji efektu uczenia się								Forma zajęć na której zachodzi weryfikacja	
EU1	test zaliczający wykład								W	
EU2	test zaliczający wykład								W	
EU3	dokumentacja zadań programistycznych, dyskusja, obserwacja pracy na zajęciach								PS	
EU4	dokumentacja zadań programistycznych, dyskusja, obserwacja pracy na zajęciach								PS	
Bilans nakładu pracy studenta (w godzinach)									Liczba godz.	
Wyliczenie	1 - Udział w wykładach - 15x1h								15	
	2 - Udział w pracowni specjalistycznej - 15x2h								30	
	3 - Praca nad projektami, przygotowanie sprawozdań								24	
	5 - Udział w konsultacjach								2	
	6 - Przygotowanie do egzaminu								15	
	7 - Obecność na egzaminie								2	
	RAZEM:								88	
Wskaźniki ilościowe									GODZINY	ECTS
Nakład pracy studenta związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela									49 (1)+(7)+(2)+(5)	1.7
Nakład pracy studenta związany z zajęciami o charakterze praktycznym									54 (3)+(2)	1.8
Literatura podstawowa	1. D. B. Kirk, W. W. Hwu, Programming massively parallel processors: a hands-on approach, 3rd edition, Elsevier/Morgan Kaufmann, 2017. 2. S. Duane, Y. Mete, CUDA for Engineers: An Introduction to High-Performance Parallel Computing, Prentice Hall, 2015.									

	3. J. C. Grossman; T. McKercher, Professional CUDA C programming, Wiley, 2014. 4. R. Farber, CUDA Application Design and Development, Elsevier 2011. 5. N. Wilt, The CUDA handbook : a comprehensive guide to GPU programming, Addison-Wesley Publishing, 2013	
Literatura uzupełniająca	1. J. Sanders, E. Kandrot, CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU, Helion, 2012. 2. J. Han, B. Sharma, Learn CUDA programming : a beginner's guide to GPU programming and parallel computing with CUDA 10.x and C/C++, Packt Publishing, 2019 3. N. Matloff, Parallel computing for data science with examples in R, C++ and CUDA, CRC/Taylor & Francis, 2015 4. R. Farber, Parallel Programming with OpenACC, Morgan Kaufmann Publishers, 2017 5. R. Robey, Y. Zamora, Parallel and high performance computing, Manning, 2021	
Jednostka realizująca	Katedra Oprogramowania	Data opracowania programu
Program opracował(a)		2021.10.27

wydrukowane w programie Świerk , © NowaSosna