

RTU studiju kurss "Paralēlā programmēšana transporta sistēmās"

31000 Būvniecības un mašīnzinību fakultāte

Vispārējā informācija

Kods	BM0935
Nosaukums	Paralēlā programmēšana transporta sistēmās
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles
Atbildīgais mācītbspēks	Māris Gailis - Doktors, Asociētais profesors
Mācītbspēks	Andrejs Potapovs - Doktors, Asociētais profesors Mihails Gorobecs - Doktors, Profesors
Apjoms daļās un kredītpunktos	2 daļas, 6.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	<p>Studiju kursā apgūst paralēlās programmēšanas principus, aprēķinu un modelēšanas un simulācijas ātrdarbībai. Šim nolūkam dalībnieki apgūst programmēšanas valodu C/C++ un Julia pamatus, pēc tam to izmantošanas prasmes palielina veikspēju lielo datu analizē transporta uzdevumos un transporta sistēmu dinamisko tehnisko procesu modelēšanā. Šo metožu realizācijai tiek izmantoti RTU HPC centra skaitļošanas resursi.</p> <p>Studiju kurss ir pielāgots kombinēto studiju metodikai, un ietver asinhronas un sinhronas studiju aktivitātes, kā arī nepieciešamos atbalsta materiālus studiju asinhronām aktivitātēm.</p> <p>Studiju kursa realizācija veicama pārsvarā ar atklāta koda brīvpieejas programmatūru. Studiju kursā izmanto izstrādes vides Spyder, Visual Studio Code (VSCode), Jupyter Lab, kā arī programmēšanas valodas Python, C/C++ un Julia ar zinātniskās skaitļošanas funkciju bibliotēkām. Izstrādes vidē Jupyter Lab ir pieejami studiju kursa mācību materiāli, vienā dokumentā apkopojot tekstu, attēlus kā arī darbojošos programmēšanas koda piemērus. Lietojot Jupyter Lab, iespējams arī izpildīt skaitļošanas uzdevumus ar RTU HPC centra superskaitļotāju. Jauna koda radīšanai studiju kursā lietota izstrādes vide Spyder, strādājot ar programmēšanas valodu Python. Savukārt izstrādes vide Visual Studio Code apgūst programmēšanas valodas C/C++ un Julia ar OpenMP un MPI funkcijām.</p> <p>Studenti apgūst studiju kursā iekļauto izstrādes rīku un programmēšanas valodu atšķirības un piemēroību dažādu uzdevumu veikšanai.</p> <p>Studenti arī apgūst augstas veikspējas skaitļošanas paņēmienus, lietojot programmēšanas valodas C/C++ un Julia un RTU HPC centra resursus. Studiju kursā paredzamo uzdevumu izpildei nepieciešams pārzināt programmēšanas valodu Python, C/C++ un Julia sintaksi un efektīvas programmēšanas paņēmienus. Tāpat apgūst un lieto šo programmēšanas valodu paralēlo funkciju bibliotēkas OpenMP un MPI, kā arī rezultātu vizualizācijas bibliotēkas. Vairākās valodas viena studiju kursā ietvaros parāda gan programmēšanas pamatprincipu līdzības, gan katras valodas priekšrocības.</p> <p>Apgūstot studiju kursu, studenti spēj izveidot digitālos datu analīzes rīkus, kas ir izprotami un izmantojami arī citiem lietotājiem, un spēj apstrādāt liela apjoma datus. Tāpat studenti spēj izstrādāt transporta sistēmu dinamisku tehnisko procesu modeļus ar paralēlās programmēšanas metodēm un risināt inženieru uzdevumus ar augstas veikspējas skaitļošanas tehnoloģijām. Studiju kursā studējošie apgūst Eiropas iedzīvotāju digitālās kompetences ietvaram (DigComp) atbilstošās augstāko, 7. un 8. līmeņu digitālās prasmes.</p>
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	<p>Studiju kursa mērķis ir attīstīt studējošo izpratni par mūsdienīgām augstas veikspējas skaitļošanas tehnoloģijām un to izmantošanu transporta jomā.</p> <p>Studiju kursa uzdevumi ir attīstīt studējošo prasmes:</p> <ul style="list-style-type: none"> - sadalīt problēmas risinājumu paralēlos apakšprocesos un programmēt tos atbilstoši augstas veikspējas skaitļošanas videi; - veikt datorizētus aprēķinus un transporta dinamisko procesu datormodelēšanu augstas veikspējas skaitļošanas vidē.
Patstāvīgais darbs, tā organizācija un uzdevumi	Ievērojot uzdotos termiņus, jāiegūst informācija no uzdota avota, jāveic pārbaudes darbi. Jāizpilda uzdotie patstāvīgie uzdevumi. Noteikti skaitļošanas uzdevumi jāsaģatavo augstas veikspējas skaitļošanas videi.
Literatūra	<p>Obligātā/Obligatory:</p> <ol style="list-style-type: none"> Allen B. Downey . Modeling and Simulation in Python: An Introduction for Scientists and Engineers, San Francisco, No Starch Press, 2023. Bjarne Stroustrup. The C++ Programming Language, 4th Edition San Francisco, Addison-Wesley Professional, 2013. Ben Lauwens, Allen B. Downey, Think Julia, New York, O'Reilly Media, Inc., 2019. <p>Papildu/Additional:</p> <ol style="list-style-type: none"> Lee Vaughan, Impractical Python Projects: Playful Programming Activities to Make You Smarter, San Francisco, No Starch Press, 2023. Rowse Samuel. Python For Data Science : what every Python developer needs to know, step by step, learn the basics of machine learning and Big Data analysis, New York, Samuel Rowse, 2022.
Nepieciešamās priekšzināšanas	Nav nepieciešamas.

Studiju kursa saturs

Saturs	Pilna un nepilna laika klātienē studijas		Nepilna laika neklātienē studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs

Inženieraprēķiniem piemērotas brīvpieejas atklātā koda izstrādes vides un programmēšanas valodas.	2	2	0	0
Izstrādes vide Jupyter Lab, teksta noformēšanas valodu Markdown un LaTeX sintakse.	4	4	0	0
Izstrādes vide Spyder.	2	2	0	0
Izstrādes vide Visual Studio Code.	2	2	0	0
Programmēšanas valodas Python priekšrocības, sintakse.	12	12	0	0
Programmēšanas valodas Julia priekšrocības, sintakse.	10	10	0	0
Programmēšanas valodas C/C++ priekšrocības, sintakse.	12	12	0	0
Augstas veiktspējas skaitļošanas (HPC) algoritmi un risināmo uzdevumu apakšprocesu paralelizācija.	4	4	0	0
Zinātniskās skaitļošanas funkciju bibliotēkas.	4	4	0	0
RTU HPC centra skaitļošanas resursi un to lietošanas paņēmieni.	2	2	0	0
OpenMP funkcijas kopatmiņas izmantošanai paralēlo aprēķinu veikšanā.	4	4	0	0
MPI funkcijas paralēlo aprēķinu veikšanai sadalītās sistēmās.	4	4	0	0
Versiju kontroles un programmas koda izvietojuma sistēma GitHub.	4	4	0	0
Programmēšanas valodu Python, C/C++ un Julia lietojums inženieru aprēķiniem, modelēšanai un rezultātu vizualizācijai.	4	4	0	0
Viedas transporta infrastruktūras datu iegūšana un lielo datu analīze ar augstas veiktspējas skaitļošanas (HPC) tehnoloģijām.	6	6	0	0
Transporta sistēmu dinamisku tehnisko procesu modelēšana ar augstas veiktspējas skaitļošanas (HPC) tehnoloģijām.	4	4	0	0
Kopā:	80	80	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Spēj izvēlēties dažādiem inženieraprēķiniem piemērotu augstas veiktspējas skaitļošanas metodi, brīvpieejas atklātā koda izstrādes vidi, programmēšanas valodu un funkciju bibliotēkas.	Pārbaudes darbi un individuālie uzdevumi, uzdevums eksāmenā.
Prot izmantot izstrādes vidi, pārzina programmēšanas valodu Python, Julia un C/C++ sintakses ar OpenMP un MPI funkcijām.	Pārbaudes darbi un individuālie uzdevumi, uzdevums eksāmenā.
Prot lietot versiju kontroles un programmas koda izvietojuma sistēmu GitHub.	Pārbaudes darbi un individuālie uzdevumi.
Prot izveidot un lietot interaktīvas piezīmju grāmatas Jupyter Lab formātā, kā arī spēj veidot formatētu tekstu lietojot Markdown un Latex sintaksi. Studiju kursa dalībnieku radītais digitālais dokuments satur gan statiskas sadaļas – tekstus, tabulas, attēlus, gan programmas kodu, kas ļauj aktīvi darboties dokumenta lietotājam, ģenerējot interaktīvas diagrammas un attēlus, un resursietilpīgu skaitļošanu veicot uz superskaitļotāja klastera (HPC) (DigComp 7. līmenis).	Pārbaudes darbi un individuālie uzdevumi, uzdevums eksāmenā.
Spēj novērtēt uzdevumam nepieciešamos superskaitļotāja resursus, kā arī prot tos lietot, lietojot Jupyter Lab un citas saskarnes.	Pārbaudes darbi un individuālie uzdevumi.
Spēj izveidot digitālos datu analīzes rīkus, kas ir izprotami un izmantojami arī citiem lietotājiem, un spēj apstrādāt liela apjoma datus, tos iegūstot no vairākiem viedās transporta infrastruktūras avotiem (DigComp 8. līmenis).	Individuālais uzdevums.
Spēj sastādīt un realizēt efektīvus skaitļošanas algoritmus programmēšanas valodās Python, C/C++ un Julia inženieraprēķiniem, datormodelēšanai un rezultātu vizualizācijai (DigComp 7. līmenis).	Individuālais uzdevums.
Spēj izstrādāt transporta sistēmu dinamisku tehnisko procesu modeļus ar paralēlās programmēšanas metodēm un risināt inženieru uzdevumus ar augstas veiktspējas skaitļošanas (HPC) tehnoloģijām (DigComp 7. līmenis).	Individuālais uzdevums.
Pārzina sistēmu vadības paralelizācijas pamatprincipus, paralēlās programmēšanas modeļus, daudzkodolu sistēmu organizāciju, pavedienu programmēšanas paņēmienus.	Pārbaudes darbi un individuālie uzdevumi, uzdevums eksāmenā.

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Pārbaudes darbi	30
Individuālie uzdevumi	30
Eksāmeni	40
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbauījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs
1.	3.0	20.0	20.0	0.0		*	
2.	3.0	20.0	20.0	0.0		*	