

Enginyeria del Rendiment

Codi: 104341
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2503758 Enginyeria de Dades	OB	2	1

La metodologia docent i l'avaluació proposades a la guia poden experimentar alguna modificació en funció de les restriccions a la presencialitat que imposin les autoritats sanitàries.

Professor/a de contacte

Nom: Juan Carlos Moure Lopez
Correu electrònic: JuanCarlos.Moure@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: espanyol (spa)
Grup íntegre en anglès: No
Grup íntegre en català: No
Grup íntegre en espanyol: No

Equip docent

Juan Carlos Moure Lopez

Prerequisits

És molt recomanable haver cursat i aprovat les assignatures de Fonaments d'Informàtica, Fonaments de Programació i Programació Avançada.

Objectius

1. Avaluar i analitzar el rendiment de l'execució d'un programa.
2. Entendre la conversió des del llenguatge font (Python/C) a l'assemblador, identificant les transformacions en el codi.
3. Determinar les dependències de dades en els llaços d'execució dels programes i fer servir estratègies per augmentar el paral·lelisme d'instruccions i de dades.
4. Comprendre el funcionament de la jerarquia de memòria i utilitzar estratègies per augmentar la localitat d'accés a les dades.

Competències

- Avaluar de manera crítica el treball realitzat.
- Buscar, seleccionar i gestionar de manera responsable la informació i el coneixement.
- Dissenyar solucions algorítmiques eficients per a problemes computacionals, implementar-les en forma de desenvolupaments de programari robust, estructurat i fàcil de mantenir, i verificar-ne la validesa.
- Planificar i gestionar el temps i els recursos disponibles.
- Que els estudiants hagin demostrat que comprenen i tenen coneixements en una àrea d'estudi que parteix de la base de l'educació secundària general, i se sol trobar a un nivell que, si bé es basa en llibres de text avançats, inclou també alguns aspectes que impliquen coneixements procedents de l'avantguarda d'aquell camp d'estudi.

Resultats d'aprenentatge

1. Analitzar el rendiment d'aplicacions, detectar colls d'ampolla i aplicar possibles optimitzacions.
2. Aplicar els principis bàsics de l'estructura i la programació dels computadors.
3. Avaluar de manera crítica el treball realitzat.
4. Buscar, seleccionar i gestionar de manera responsable la informació i el coneixement.
5. Implementar en un llenguatge de programació problemes algorítmics de dificultat mitjana.
6. Planificar i gestionar el temps i els recursos disponibles.
7. Que els estudiants hagin demostrat que comprenen i tenen coneixements en una àrea d'estudi que parteix de la base de l'educació secundària general, i se sol trobar a un nivell que, si bé es basa en llibres de text avançats, inclou també alguns aspectes que impliquen coneixements procedents de l'avantguarda d'aquell camp d'estudi.

Continguts

1. Avaluació del Rendiment: eines per monitoritzar i instrumentar l'execució d'un programa.
2. Anàlisi del Rendiment: complexitat computacional, factors determinats per l'arquitectura del processador i mètriques de rendiment.
3. Paral·lelisme en l'execució d'Instruccions: Pipeline i Vectorització.
4. Anàlisi de Dependències de dades en els llaços d'execució.
5. Jerarquia de Memòria i Localitat d'accés a les dades: efecte en el rendiment.
6. Estratègies algorítmiques d'optimització: augmentar el paral·lelisme i la localitat d'accés a les dades.

Metodologia

Teoria: s'exposaran els coneixements propis de l'assignatura. Es descriuran els conceptes bàsics i s'il·lustraran amb exemples pràctics. Es ressaltaran els problemes d'aprenentatge més importants i es mostrarà com completar i aprofundir en els continguts. Es discutiran casos pràctics i el professor detectarà els problemes de comprensió i raonament més comuns i els resoldrà per a tots els estudiants.

Problemes en Laboratori: es faran activitats cooperatives de resolució de problemes i de casos pràctics senzills, que serviran de suport a la teoria. Després d'un treball individual previ, els i les alumnes faran una posada en comú en grup i resoldran els seus dubtes. El professor detectarà els problemes de comprensió i raonament més comuns i els resoldrà per grups o per a tot l'alumnat. Es promou la capacitat d'anàlisi i síntesi, el raonament crític i la capacitat de resolució de problemes. Durant la sessió les i els alumnes hauran d'informar al professor sobre els seus avenços i els problemes que es puguin trobar, i lliuraran un document final amb els resultats i reflexions sobre les seves activitats.

Problemes d'Optimització: durant tot el curs, es proposaran exercicis per transformar petits trossos de codi perquè s'executin en menys temps. Els estudiants han d'aplicar els coneixements, metodologies i estratègies apreses i presentar el resultat argumentant les raons de les millores en el rendiment. S'espera que els estudiants siguin proactius i dinàmics, i la recerca autònoma dels coneixements necessaris.

Nota: es reservaran 15 minuts d'una classe, dins del calendari establert pel centre/titulació, per a la complementació per part de l'alumnat de les enquestes d'avaluació de l'actuació del professorat i d'avaluació de l'assignatura/mòdul.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classes de Teoria	26	1,04	4, 7
Resolució de problemes pràctics al laboratori	26	1,04	1, 2, 3, 5, 6

Tipus: Supervisades

Optimització del Rendiment de fragments de codi	26	1,04	1, 2, 3, 4, 5, 6
Tipus: Autònomes			
Estudi autònom	30	1,2	4, 6
Preparació prèvia i documentació del treball al laboratori	40	1,6	1, 3, 4, 6, 7

Avaluació

Les dates d'avaluació continuada i lliurament de treballs es publicaran a la plataforma Moodle de la UAB (Campus Virtual) i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. Sempre s'informarà a la plataforma Moodle sobre aquests canvis, ja que s'entén que és la plataforma habitual d'intercanvi d'informació entre professor i estudiants.

Activitats d'avaluació

Activitat A: resolució de problemes pràctics al laboratori; es realitza en grup; suposa un 50% de la nota final; cal treure una nota mínima de 5 punts sobre 10; i no es pot recuperar.

Activitat B: lliurament d'un informe de l'activitat A; es realitza individualment; suposa un 15% de la nota final; no cal treure una nota mínima; i no es pot recuperar.

Activitat C: assistència i participació activa a classe; es realitza individualment; suposa un 5% de la nota final; no cal treure nota mínima; i no es pot recuperar.

Activitat D: examen escrit de teoria i problemes; es realitza individualment i sense apunts; suposa un 30% de la nota final; cal treure una nota mínima de 5 punts sobre 10; i es pot recuperar.

A part de les activitats d'avaluació descrites anteriorment, al llarg del curs es podran proposar altres activitats de caire voluntari que podran contribuir a la nota final de l'assignatura

Programa d'activitats d'avaluació

Les activitats A, B i C es realitzen durant tot el curs. L'activitat D es realitza a final de curs, en les dates especificades per la coordinació.

Procés de recuperació

L'estudiant i la estudiant es pot presentar a la recuperació sempre que s'hagi presentat a les activitats A, C i D. Només es podrà recuperar l'activitat D, que representa el 30% de la nota final.

Procediment de revisió

Totes les activitats d'avaluació es podran revisar a un lloc, data i hora que es publicarà amb anterioritat. Es podran fer reclamacions sobre la nota de l'activitat, que seran avaluades pel professorat responsable de l'assignatura. Si l'estudiant no es presenta a aquesta revisió, no es revisarà posteriorment aquesta activitat.

Qualificacions

S'atorgarà una qualificació de matrícula d'honor (MH) als estudiants que assoleixin una nota de qualificació final igual o superior a 9,00, fins a un total del 5% dels estudiants matriculats, prioritzant la part de la nota que correspon a avaluacions individuals (activitats B, C i D).

Si l'alumne o l'alumna no assoleix la nota mínima en alguna de les activitats d'avaluació, la qualificació numèrica final serà el mínim entre 4,5 punts i la suma ponderada de les notes de totes les activitats.

Un estudiant es considerarà no avaluable (NA) si no s'ha presentat a cap de les activitats A, B i D.

Irregularitats per part dels estudiants, còpia i plagi

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, es qualificaran amb un zero les irregularitats comeses per un estudiant que puguin conduir a una variació de la qualificació d'un acte d'avaluació. Per tant, la còpia, el plagi, l'engany, deixar copiar, etc. en qualsevol de les activitats d'avaluació implicarà suspendre-la amb un zero. Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs. En cas de no superar l'assignatura degut a que l'estudiant ha comès irregularitats en un acte

d'avaluació, la nota numèrica de l'expedient serà el valor menor entre 3.0 i la mitjana ponderada de les notes (i per tant no serà possible l'aprovat per compensació).

Avaluació dels estudiants repetidors

No hi ha cap tracte especial pels estudiants repetidors.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Examen escrit individual de teoria i problemes	30%	2	0,08	1, 3, 7
Informe individual de la resolució de problemes pràctics al laboratori	15%	0	0	1, 3, 4, 7
Participació a classe/laboratori	5%	0	0	1, 3, 4, 6, 7
Treball en grup per resoldre problemes pràctics al laboratori + Respostes + Informes	50%	0	0	1, 2, 4, 5, 6, 7

Bibliografia

Computer Architecture: A Quantitative Approach. 6th Edition

(https://cataleg.uab.cat/iii/encore/record/C__Rb2034006)

John Hennessy and David Patterson, Morgan Kaufmann (Elsevier) 2019 (Cap. 1-3)

Computer Systems: A Programmer's Perspective. 3rd Edition (North-American)

(https://csuc-uab.primo.exlibrisgroup.com/permalink/34CSUC_UAB/avjcib/alma991006392489706709)

Randall E. Bryant and David R. O'Hallaron, Addison Wesley, 2015 (Cap. 5 i 6)

Computer Organization and Design RISC-V Edition: The Hardware/Software Interface.

David Patterson and John Hennessy, Morgan Kaufman (Elsevier), 2017 (Cap. 4 i 5)

High Performance Python: practical performant programming for human.

Micha Gorelick and Ian Ozsvald, O'Reilly, 2014

Cython: a guide for Python programmers.

Kurt W. Smith, O'Reilly, 2015

Python High Performance programming. Boost the performance of your Python programs using advanced techniques

Gabriele Lanaro, Packt Publishing, 2013

Programari

Cap