

## Guia docent

### Identificació de l'assignatura

<b>Assignatura / Grup</b>	21743 - Compiladors II / 1
<b>Titulació</b>	Grau d'Enginyeria Informàtica (Pla 2010) - Tercer curs Grau d'Enginyeria Informàtica (Pla 2014) - Tercer curs
<b>Crèdits</b>	6
<b>Període d'impartició</b>	Segon semestre
<b>Idioma d'impartició</b>	Català

### Professors

Professor/a	Horari d'atenció als alumnes					
	Hora d'inici	Hora de fi	Dia	Data d'inici	Data de fi	Despatx / Edifici
Pedro Antonio Palmer Rodríguez (Responsable) <a href="mailto:pere.palmer@uib.es">pere.palmer@uib.es</a>	16:30	17:30	Divendres	10/09/2018	25/01/2019	Anselm Turmeda D142
	10:30	11:30	Divendres	10/09/2018	28/06/2019	Anselm Turmeda D142

### Contextualització

Els compiladors són programes que prenen com a dades codi escrit en llenguatges de programació d'alt nivell i el converteixen a versions equivalents que siguin fàcilment executables per una màquina. L'assignatura té per objectiu explicar les dificultats d'aquest procés i les tècniques que existeixen per a resoldre-les. Aquestes darreres fan ús combinat del bagatge adquirit en diferents assignatures prèvies, sobretot, teoria d'autòmats, estructures de dades, llenguatges de programació i estructura de computadors.

El coneixement de l'estructura i funcionament d'un compilador aporta diferents avantatges a l'estudiant i al professional:

En primer lloc articula i dona sentit a molts dels coneixements que haurà vist en assignatures anteriors (teoria d'autòmats, estructures de dades, arquitectura de computadors, teoria de grafs, etc.) i hi trobarà la manera de combinar-los per resoldre un problema de tanta importància pràctica com és la traducció d'un llenguatge de programació en codi màquina.

En segon lloc, per un programador, la comprensió de com un compilador tractarà el que el codi que ell escriu l'ajudarà a ser conscient de les implicacions que tenen les seves decisions en l'eficiència del codi i l'ajudaran a construir programes de major qualitat.

Finalment, la pràctica de l'assignatura constitueix un excel·lent exercici de programació en el qual l'estudiant s'enfronta per primera vegada a la dificultat d'escriure un programa gran.

En l'assignatura de Compiladors I ja s'han abordat els aspectes frontals d'un compilador, és a dir els que depenen del llenguatge font: l'anàlisi lèxica, l'anàlisi sintàctica i la comprovació de tipus. En l'assignatura de Compiladors II s'aborden els aspectes dorsals: generació de codi intermedi, optimització de codi, gestió de memòria i generació de codi assemblador.

Acabats el dos cursos, l'estudiant estarà en condicions de construir un compilador i d'entendre els detalls d'implementació de les construccions pròpies dels llenguatges d'alt nivell. Les tècniques que haurà conegut

## Guia docent

serveixen també per a la construcció de formatejadors de text (LaTeX i similars), llenguatges gràfics (Postscript, GIF, JPEG), i llenguatges de descripció de hardware (VDHL, Verilog).

### Requisits

L'estudiant que cursi aquesta assignatura haurà de tenir sòlids coneixements de programació, estructures de dades, teoria d'autòmats, llenguatges de programació i estructura de computadors. Naturalment, també haurà d'haver cursat l'assignatura de Compiladors I

#### Essencials

L'estudiant que cursi aquesta assignatura haurà de tenir sòlids coneixements de programació, estructures de dades, teoria d'autòmats, llenguatges de programació i estructura de computadors.

#### Essencials

- 1) Programació I
- 2) Programació II
- 3) Algoritmia
- 4) Estructures de dades
- 5) Teoria de la computació
- 6) Estructura de computadors I
- 7) Estructura de computadors II
- 8) Compiladors I

#### Recomanables

Seria recomanable que haguessin cursat l'assignatura de Llenguatges de programació.

### Competències

#### Específiques

- \* CI302 Capacitat per conèixer els fonaments teòrics dels llenguatges de programació i les tècniques de processament lèxic, sintàctic i semàntic associades, i saber aplicar-les per a la creació, el disseny i el processament de llenguatges. .

#### Genèriques

- \* CTR01 Capacitat d'anàlisi i síntesi, d'organització, de planificació i de presa de decisions. .
- \* CTR02 Capacitat d'anàlisi crítica i de proposta i aplicació de noves solucions. .
- \* CTR03 Capacitat per adquirir de forma autònoma nous coneixements. .
- \* CTR04 Capacitat per a la recerca de recursos i de gestió de la informació en l'àmbit de la informàtica .
- \* CTR07 Capacitat per comunicar conceptes propis de la informàtica de manera oral i escrita en diferents àmbits d'actuació. .



## Guia docent

### Bàsiques

- \* Podeu consultar les competències bàsiques que l'estudiant ha d'haver assolit en acabar el grau a l'adreça següent: [http://estudis.uib.cat/ca/grau/comp\\_basiques/](http://estudis.uib.cat/ca/grau/comp_basiques/)

### Continguts

Essent Compiladors II la continuació natural de Compiladors I, els temes s'han numerat correlativament respecte als temes d'aquesta.

#### Continguts temàtics

TEMA VI. Anàlisi semàntica III: Generació de codi intermedi

TEMA VI: ANÀLISI SEMÀNTICA III: GENERACIÓ DE CODI INTERMEDI

26. Codi intermedi (llenguatges imperatius). Conveniència i característiques desitjables. Codi de tres adreces. Representació. Taula de variables i taula de subprogrames.

27. Generació de codi intermedi: expressions aritmètiques. Alternatives respecte al codi generat: variables temporals o gestió de pila. Llurs implicacions ulteriors en el procés d'optimització. Atributs i rutines semàntiques en cada cas.

28. Generació de codi intermedi: referències a matrius i tuples. Índexos i càlcul de posicionament relatiu. Atributs i rutines semàntiques associades.

29. Generació de codi intermedi: expressions lògiques. Operadors lògics com a estructures de control del flux d'execució. Tècniques d'etiquetat i d'incorporació retroactiva d'adreces. Atributs i rutines semàntiques associades en cada cas.

30. Generació de codi intermedi: estructures de control. Sentències alternativa i repetitiva. Atributs i rutines semàntiques associades en tècniques d'etiquetat i en tècniques d'incorporació retroactiva d'adreces.

TEMA VII. Anàlisi semàntica IV: Gestió de memòria

TEMA VII: ANÀLISI SEMÀNTICA IV: GESTIÓ DE MEMÒRIA

31. Gestió de memòria: variables. Memòria estàtica i dinàmica. Blocs d'activació. Llur implicació en la implementació de la recursivitat. L'accés a variables no locals. Variables locals de dimensió variable.

32. Gestió de memòria: paràmetres. Pas per valor. Pas per referència. Pas per valor-resultat.

33. Gestió de memòria: variables dinàmiques. Ubicació dinàmica de blocs de memòria. Generació de codi per al desencadenament de processos de recollida de residus.

34. Gestió de memòria: subprogrames: operacions associades a lacrida, preàmbul i retorn d'invocacions a subprogrames.

TEMA VIII. Optimització de codi

TEMA VIII: OPTIMITZACIÓ DE CODI

35. Visió general del procés d'optimització. Les limitacions del codi generat per traduccions guiades per la sintaxi: necessitat del procés d'optimització. Independència d'aquest procés respecte al llenguatge font i a la màquina de destí. Noció de les optimitzacions bàsiques:

## Guia docent

eliminació d'expressions comunes, propagació de còpies, eliminació de codi mort, extracció d'invariants en bucles, variables d'inducció.

36. Optimitzacions de finestra. Reducció de brancaments. Simplificacions algebraiques. Reemplaçament de components escalars per variables. Patrons reduïbles habituals.

37. Grafs de flux: conceptes generals. Concepte i objectius. Representació. Algoritme de construcció.

38. Grafs de flux: anàlisi estructural. Nodes dominants. Bucle natural. Preencapçaladors. Imbricació de bucles. Grafs reduïbles. Grafs jerarquitcats.

39. Grafs de flux: anàlisi de flux de dades. Concepte i objectius. Conjunts d'entrada, de sortida, generats i emmascarats. Equacions de flux. Tipologia de casos: les quatre classes de problemes. Algoritmes generals de resolució per a cadascuna de les classes. Algoritmes de resolució en el cas de grafs jerarquitcats. El problema de l'ajustament dels resultats després de canvis en el codi. Càlcul de variables vives, definicions accessibles, cadenes definició-ús, cadenes ús definició, expressions disponibles, expressions molt ocupades.

40. Optimització global. Propagació de còpies. Eliminació d'expressions comunes. Extracció d'invariants en bucles. Variables d'inducció. Eliminació de codi mort i inaccessible. Actuació combinada dels diferents mecanismes d'optimització: ordre d'aplicació i reiteració.

41. Optimització local. Graf dirigit acíclic de precedències d'operacions dins un bloc bàsic. Algoritmes de construcció. Heurístics d'ordenació topològica en relació a l'algoritme bàsic de generació de codi objecte. Algoritme d'ordenació topològica òptima en el cas d'arbres. Algoritme de generació de codi intermedi a partir de l'ordenació topològica resultant.

### TEMA IX. Generació de codi assemblador

#### TEMA IX. GENERACIÓ DE CODI OBJECTE

42. Codi objecte. Característiques generals. Registres. Modes d'adreçament. Repertori d'instruccions. Cicles de memòria. Objectius de qualitat en el codi generat.

43. Generació de codi: mecanisme elemental. Traducció instrucció a instrucció. Ineficiència del codi resultant. Reducció de força.

44. Generació de codi: algoritme bàsic. Descripció del contingut dels registres. Descripció de la ubicació dels valors de les variables. Generació de codi minimitzant el nombre de transferències memòria-registre. Heurístics en l'assignació de registres.

45. Millores a l'algoritme bàsic de generació de codi. Registres especialitzats. Assignació global de registres en bucles més interns. Estratègia de recompte d'usos. Assignació de registres per colorejat de grafs.

46. Generadors de generadors de codi. Modes d'utilització. Principis bàsics de funcionament. Idiomes màquina.

### TEMA X. Generació de codi per llenguatges funcionals

#### TEMA X. GENERACIÓ DE CODI EN LLENGUATGES FUNCIONALS

47. Naturalesa dels llenguatges funcionals. Transparència referencial. Manipulació de funcions com a objectes. Llistes. Unificació. Modes d'avaluació: àvida i peresosa. Avaluació peresosa i definició d'estructures de dades infinites.

48. Codi intermedi per a llenguatges funcionals. Fonament matemàtic: el lambda-càlcul. El lambda-càlcul com a codi intermedi: operacions incorporades. Reduccions lambda i beta. Redex. Ordres de reducció normal i aplicatiu. Formes normals. El problema de l'acabament. Reduccions lambda i conflictes de noms. Formes normals de cap feble. Relació entre ordres de



## Guia docent

reducció de lambda-càlcul i modes d'avaluació dels llenguatges funcionals. La recursivitat en el lambda-càlcul: l'operador de punt fix Y.

49. Traducció de llenguatges funcionals a lambda-càlcul. Codi associat al mecanisme d'unificació. Criteris de resolució d'ambigüitats. Codi per a l'establiment de vincles.

50. Interpretació de lambda-càlcul. Interpretació aplicativa: la màquina SECD. Interpretació normal: reducció de grafs. Discussió sobre l'eficiència de l'execució interpretada.

51. Lògica combinatòria com a segon nivell de codi intermedi. Els combinadors SK. Traducció d'expressions lambda-càlcul a expressions de combinadors. El problema de l'eficiència. Extensió del repertori de combinadors: els combinadors B i C. Algorisme de traducció optimitzada. Interpretació d'expressions de combinadors mitjançant reducció de grafs. Representació eficient del combinador Y. Cadenes directores.

52. Supercombinadors. Definició de combinadors d'alt nivell a la mida del programa per elevació de lambdas.

53. Generació de codi objecte per a l'avaluació àvida. El codi imperatiu intermedi FPM. Traducció de supercombinadors a FPM.

54. Generació de codi objecte per a l'avaluació àvida. El codi imperatiu intermedi màquina-G. Conversió d'expressions de supercombinadors en un manipulador imperatiu de grafs.

## Metodologia docent

Activitats de treball presencial (2,4 crèdits, 60 hores)

Modalitat	Nom	Tip. agr.	Descripció	Hores
Classes teòriques	Teoria	Grup gran (G)	S'hi explica la manera de resoldre els diferents aspectes del procés de compilació que es tracten en el temari i part de la teoria que hi dona suport. Només part, perquè bona part dels aspectes teòrics ja s'han explicat en assignatures precedents.  Un cop desenvolupats els aspectes teòrics, es dediquen unes setmanes de classes a la resolució magistral de problemes que els estudiants haurien d'haver intentat resoldre prèviament pel seu compte.	60

A començament del semestre hi haurà a disposició dels estudiants el cronograma de l'assignatura a través de la plataforma UIBdigital. Aquest cronograma inclourà almenys les dates en què es faran les proves d'avaluació contínua i les dates de lliurament dels treballs. A més, el professor o la professora informará els estudiants si el pla de treball de l'assignatura es durà a terme a través del cronograma o per una altra via, inclosa la plataforma Aula Digital.

Activitats de treball no presencial (3,6 crèdits, 90 hores)

## Guia docent

Modalitat	Nom	Descripció	Hores
Estudi i treball autònom individual o en grup	Pràctica	<p>Els estudiants, en grups petits de entre dos i quatre membres, hauràn de desenvolupar un compilador. Durant l'assignatura Compiladors I, ja hauran resolt la part frontal (anàlisi lèxica, sintàctica, la taula de noms i la taula de símbols).</p> <p>En l'assignatura de Compiladors II hauran de reprendre el treball efectuat fins a arribar a la generació de codi assemblador.</p> <p>A mitjans del quadrimestre hi haurà una sessió de control destinada al seguiment de la feina efectuada i, sobretot, de correcció a temps de decisions que puguin perjudicar el desenvolupament posterior del treball.</p>	90

### Riscs específics i mesures de protecció

Les activitats d'aprenentatge d'aquesta assignatura no comporten riscos específics per a la seguretat i salut dels alumnes i, per tant, no cal adoptar mesures de protecció especials.

### Avaluació de l'aprenentatge dels estudiants

#### Frau en elements d'avaluació

D'acord amb l'article 33 del Reglament acadèmic, "amb independència del procediment disciplinari que es pugui seguir contra l'estudiant infractor, la realització demostradorament fraudulenta d'algun dels elements d'avaluació inclosos en guies docents de les assignatures comportarà, a criteri del professor, una menysvaloració en la seva qualificació que pot suposar la qualificació de «suspens 0» a l'avaluació anual de l'assignatura".

#### Teoria

Modalitat	Classes teòriques
Tècnica	Proves objectives ( <b>recuperable</b> )
Descripció	S'hi explica la manera de resoldre els diferents aspectes del procés de compilació que es tracten en el temari i part de la teoria que hi dona suport. Només part, perquè bona part dels aspectes teòrics ja s'han explicat en assignatures precedents. Un cop desenvolupats els aspectes teòrics, es dediquen unes setmanes de classes a la resolució magistral de problemes que els estudiants haurien d'haver intentat resoldre prèviament pel seu compte.
Criteris d'avaluació	És imprescindible aprovar l'examen per fer mitja. Competències: CI302, CTR01, CTR02

Percentatge de la qualificació final: 50%

#### Pràctica

Modalitat	Estudi i treball autònom individual o en grup
Tècnica	Proves orals ( <b>recuperable</b> )
Descripció	Els estudiants, en grups petits de entre dos i quatre membres, hauràn de desenvolupar un compilador. Durant l'assignatura Compiladors I, ja hauran resolt la part frontal (anàlisi lèxica, sintàctica, la taula de noms i la



## Guia docent

taula de símbols). En l'assignatura de Compiladors II hauran de reprendre el treball efectuat fins a arribar a la generació de codi assemblador. A mitjans del quadrimestre hi haurà una sessió de control destinada al seguiment de la feina efectuada i, sobretot, de correcció a temps de decisions que puguin perjudicar el desenvolupament posterior del treball.

criteris d'avaluació      Competències: CI302,CTR01,CTR02, CTR03,CTR04,CTR07

Percentatge de la qualificació final: 50%

## Recursos, bibliografia i documentació complementària

### Bibliografia bàsica

- Aho, A. V., M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, Addison-Wesley, 2006. Sens cap mena de dubte, les edicions precedents d'aquest llibre han estat el text més universalment usat a les universitats per aquesta assignatura. El tractament de la part frontal del compilador és excel·lent. En canvi, la part d'optimització i generació de codi assemblador queda descrita de manera molt més imprecisa. Tampoc tracta la generació de codi per a llenguatges funcionals.
- Muchnick, S., *Advanced Compiler Design and Implementation*, Morgan Kaufman, 1997. És un llibre exhaustiu en matèria d'optimització de codi per a llenguatges imperatius. Dissortadament, la seva lectura és molt feixuga degut a l'elecció d'una notació pròpia, i excessivament críptica, per a la descripció dels algoritmes. Les explicacions no sempre són clares i, ocasionalment, errònies.
- Field, A. J. and P. G. Harrison, *Functional Programming*, Addison-Wesley, 1987. És un llibre molt bellament escrit que, de manera extraordinàriament clara, explica en profunditat la generació de codi per a llenguatges funcionals.

### Bibliografia complementària

- Zima, H. and B. Chapman, *Supercompilers for Parallel and Vector Computers*, Addison-Wesley, 1991. Focalitzat en l'extracció de paral·lelisme potencial a partir del codi intermedi generat per la part frontal d'un compilador per a llenguatges imperatius convencionals, cara a la generació de codi per a arquitectures de paral·lelisme intensiu.

