

Compiladors

Codi: 102782

Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	3	2
2502441 Enginyeria Informàtica	OT	4	2

Professor/a de contacte

Nom: Francisco Javier Sánchez Pujadas

Correu electrònic: Javier.Sanchez.Pujadas@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: català (cat)

Grup íntegre en anglès: No

Grup íntegre en català: Sí

Grup íntegre en espanyol: No

Equip docent

Jorge Bernal del Nozal

Prerequisits

S'aconsella haver cursat les assignatures:

- Fonaments d'Informàtica
- Metodologia de la Programació
- Laboratori de Programació

Coneixements que podran ser útils per aquesta assignatura són:

- Els coneixements sobre llenguatges de programació imperatius i orientats a objecte com C/C++ facilitaràn la comprensió dels exemples pràctics donats a teoria.
- Per la part de generació de codi seran útils els coneixements sobre assamblador.
- Els coneixements sobre recursivitat de l'assignatura Anàlisi i Disseny d'Algorismes ajudaran a comprendre el funcionament de l'anàlisi sintàctic.

Objectius

Coneixements:

- Introduir els conceptes i estructures fonamentals d'un compilador de llenguatges imperatius.
- Millorar l'ús i comprensió dels compiladors (eficiència, límits del llenguatge, etc.), i dels llenguatges de programació.
- Implementació pràctica de parsers simples.
 - Reconeixadors de fitxers de configuració amb paràmetres complexos.
 - Flexibilització de diàlegs amb l'usuari.
- Resoldre problemes utilitzant tècniques sintàctiques i semàntiques.
 - Definició i implementació de gramàtiques per resoldre problemes.

- Decisió entre implementar un compilador sencer o utilitzar una eina per generar compiladors en funció del tipus de problema a resoldre.
- Implementar algorismes complexos.
- Estratègies de test de programes complexos.
- Implementació d'algorismes recursius.
- Ús d'eines que generen codi.
- Modularització en subproblemes fortament acoblats.
- Adaptació i utilització d'un programa ja fet.

Habilitats:

- Millorar la qualitat dels programes creats gràcies al major coneixement del funcionament dels compiladors.
- Utilització correcta de les estructures dels llenguatges de programació per obtenir programes més òptims.
- Resolució de problemes d'una forta complexitat aplicant els principis de modularitat i recursivitat.
- Utilització d'eines de programació automàtica.

Competències

Enginyeria Informàtica

- Adquirir hàbits de pensament.
- Adquirir hàbits de treball personal.
- Capacitat per a 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, disseny i processament de llenguatges.
- Capacitat per concebre, desenvolupar i mantenir sistemes, serveis i aplicacions informàtiques emprant els mètodes de l'enginyeria del software com a instrument per a assegurar-ne la qualitat.

Resultats d'aprenentatge

1. Adaptar els coneixements d'anàlisi lèxic, sintàctic i semàntic a la generació de codi en un compilador bàsic.
2. Aplicar els coneixements d'anàlisi lèxic, sintàctic i semàntic a la generació de codi en un compilador bàsic.
3. Conèixer i comprendre els processos d'anàlisi lèxic, sintàctic i semàntic dels llenguatges de programació i analitzar les diferents alternatives de cadascun d'ells.
4. Desenvolupar un mode de pensament i raonament crítics.
5. Gestionar el temps i els recursos disponibles. Treballar de manera organitzada.
6. Prendre decisions pròpies.
7. Prevenir i solucionar problemes.

Continguts

I. Introducció. Conceptes bàsics.

1. Introducció als compiladors.
2. Estratègies de construcció d'un compilador.
3. Definició de compilador i les seves parts.
4. Presentació dels següents temes del curs.

II. Anàlisi lexicogràfica: SCANNER.

1. Definició d'anàlisi lexicogràfica
2. Expressions regulars
3. Implementació d'un analitzador lexicogràfic (Scanner)
4. Anàlisi lexicogràfica.

III. Gramàtiques i anàlisi sintàctica: PARSER.

1. Definició de gramàtiques lliures de context.
2. Notacions per definir gramàtiques.
3. Característiques de les gramàtiques i els seus Llenguatges de Programació associats.
4. Gramàtiques LL(1).
5. Parsers LL(1).
6. Recuperació d'errors en Parsers LL(1).
7. Parsers LR, SLR, LALR.
8. Consideracions pràctiques.

IV. Anàlisi semàntica.

1. Traducció dirigida per la Sintaxi.
2. Taula de Símbols.
3. Anàlisi semàntica de declaracions.
4. Anàlisi semàntica d'expressions.

V. Organització de la memòria en temps d'execució.

1. Ambient d'execució i assignació estàtica de memòria.
2. Assignació de la pila.
3. Heap.

VI. Generació de codi.

1. Màquina abstracta.
2. Generació de codi d'expressions.
3. Procés de referències a estructures de dades.
4. Generació de codi de funcions.
5. Generació de codi d'estructures de control.

Metodologia

Aquesta assignatura té un vincle molt fort entre el que és la teoria, explicació de com funcionen i es poden crear compiladors, i la pràctica o els problemes, com es crea realment un compilador. Durant les classes de teoria s'expliquen conceptes i tècniques que són necessaris per poder resoldre la pràctica. Els conceptes de pràctica i problemes estan relacionats, atès que l'alumne dissenyarà una solució per resoldre un problema i la s'implementarà a la pràctica. D'aquesta manera, el procés d'aprenentatge de l'alumne es basarà en aquests tres tipus d'activitats: classes de teoria, seminaris o explicacions de com aplicar els conceptes teòrics a la pràctica i les classes pràctiques o de problemes en què els alumnes presentaran i discutiran els seus solucions.

Sessions de classe unificades

No es fa diferència entre classes de teoria, problemes i pràctiques. D'aquesta manera es comença la sessió explicant els conceptes teòrics com en una classe de teoria. Després s'apliquen aquests conceptes en problemes o pràctiques. Així la sessió acaba com una classe de problemes o pràctiques.

Classes de teoria

Les classes de teoria serviran per introduir els conceptes que es detallen en el temari de l'assignatura. Es basaran en les explicacions del professor, que s'ajudarà de material complementari en forma de fotocòpies o transparències. L'alumne haurà de completar les classes de teoria amb les hores d'estudi, que han de servir per acabar d'entendre els continguts de l'assignatura i poder preparar correctament les sessions de pràctiques. Cal tenir present que les classes de teoria presenten una continuïtat al llarg del curs, de manera que per poder seguir correctament una classe cal haver assimilat el que s'ha explicat en les classes anteriors.

Classes de problemes

Les classes de problemes es dedicaran bàsicament a l'explicació detallada de com aplicar els conceptes teòrics de manera pràctica, i serviran de pont entre les classes teòriques i les pràctiques. Els professors plantejaran problemes i han d'indicar com es resolen preparant als alumnes per resoldre les pràctiques. És molt important tenir assimilats els conceptes teòrics previs per poder seguir correctament les classes de problemes.

Abans de cada sessió de problemes el professor proposarà una llista d'exercicis que els alumnes hauran d'intentar resoldre prèviament per poder presentar dubtes a l'hora de classe. En algunes sessions es prepararan o discutiran problemes comuns sorgits en les pràctiques.

Pràctica

La pràctica serà fer modificacions sobre el compilador explicat a teoria. La pràctica es realitzarà en grups de 1, 2 o 3 persones. Les sessions de pràctiques estaran dividides en 3 blocs: Anàlisi sintàctica, Anàlisi semàntica i Generació de codi. Cada bloc tindrà un o més entregues. L'alumne podrà provar cada lliurament amb un autotest format per les proves que ha de passar la pràctica per a aprovar. Després el professor passarà un test més complet que serà la base per obtenir la nota de la pràctica de grup. La pràctica es podrà provar amb un corrector web totes les vegades que l'alumne necessiti abans de la data de lliurament. A més, es podrà avaluar el lliurament, i els coneixements obtinguts per cada alumne del grup amb una petita prova oral o escrita per la nota individual.

Competències transversals: Les competències transversals seran treballades i avaluades en diversos moments al llarg del curs. Alguns exemples de com es treballen són els següents:

- T01.01 - Desenvolupar un mode de pensament i raonament crítics:
 A les classes de teoria s'expliquen exemples de llenguatges i compiladors existents dels que es valora els pros, contres i errors de disseny.
 A les classes de problemes, els alumnes han de crear un model abstracte del problema per poder-lo resoldre.
 A la pràctica l'alumnat ha de deduir quines restriccions semàntiques ha de comprovar el compilador.
- T02.03 - Gestionar el temps i els recursos disponibles. Treballar de manera organitzada:
 A les pràctiques, l'alumnat ha de decidir en quin ordre resol els diversos problemes de la pràctica i quina és la seva estratègia per encarar-los.
- T02.04 - Prevenir i solucionar problemes:
 A la pràctica hi ha fortes dependències entre les diferents etapes de la compilació. L'alumnat ha de prendre decisions sobre què fa a les primeres etapes per assegurar que les següents no es compliquin massa.
- T02.05 - Prendre decisions pròpies:
 Tan a problemes com a les pràctiques, l'alumnat ha de prendre decisions sobre com resol els diversos problemes que apareixen.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classes unificades (teoria, problemes i pràctiques)	46	1,84	1, 2, 3, 4, 5, 6, 7
Tipus: Autònomes			
Estudi de la matèria impartida en classes teòriques	36	1,44	1, 2, 3, 4
Preparació i implementació de la pràctica	47	1,88	1, 2, 3, 4, 5, 6, 7
Preparació i resolució de problemes	12	0,48	1, 2, 3, 4, 5, 6, 7

Avaluació

Criteris i indicadors d'avaluació:

- Comprensió dels conceptes teòrics de l'assignatura.
- Utilització correcta de les principals eines utilitzades per construir un compilador.
- Aplicació dels principis generals de disseny en la resolució de problemes complexos.
- Testejar problemes complexos.

Activitats i instruments d'avaluació:

- Nota Pràctiques: En aquest apartat hi ha una nota de grup i una individual:
 - Nota de grup: Correspon a la nota obtinguda als lliuraments per grups de la pràctica.
 - Nota individual: És la puntuació obtinguda a l'examen de pràctiques.
- Nota Teoria: correspon a l'examen final o examen/s parcial/s de teoria sobre l'assignatura. S'ha/n d'aprovar els examen/s parcial/s per tal d'eliminar matèria a l'examen final.

La nota final de l'assignatura s'obté combinant l'avaluació d'aquestes dues activitats de la manera següent:

$$\text{Nota Final} = (0.5 * \text{Nota Pràctiques}) + (0.5 * \text{Nota Teoria})$$

$$\text{Nota Pràctiques} = (0.2 * \text{Nota Individual}) + (0.8 * \text{Nota Grup})$$

$$\text{Nota Individual} = \text{Examen de pràctiques.}$$

$$\text{Nota Grup} = (0.2 * \text{Grup A. Sintàctic}) + (0.3 * \text{Grup A. Semàntic}) + (0.5 * \text{Grup Generació de Codi})$$

Les notes de grup d'anàlisi sintàctica, semàntica i generació de codi poden correspondre a una mitjana ponderada de varies entregues.

Convalidació de pràctiques: No es convaliden pràctiques d'anys anteriors.

Recuperació de pràctiques: En el cas d'haver suspès un lliurament de grup, es podrà recuperar en els següents lliuraments de la pràctica. La nota serà $0.75 * (\max(\text{nota recuperació}, \text{nota lliurament suspès}) - \text{nota lliurament suspès}) + \text{nota lliurament suspès}$. En el cas de suspendre l'examen de pràctiques, l'alumne s'haurà de presentar a un examen de recuperació de pràctiques el mateix dia de l'examen final.

Condicions per aprovar l'assignatura:

- Nota Final ≥ 5
- Nota pràctica ≥ 5 i haver aprovat cadascun dels lliuraments individuals i de grup.
- Nota teoria ≥ 5 i haver aprovat per separat les parts que tingui l'examen de teoria.

Condicions pel no avaluable:

- No tenir cap part de l'assignatura suspesa.

Condicions pel suspens:

- No arribar a una nota mitjana superior o igual a 5.
- Suspendre alguna de les activitats d'avaluació de l'assignatura encara que la mitjana superi el 5. En aquest cas, la nota serà la nota mínima obtinguda d'alguna de les parts (exàmens o pràctiques).

Condicions per la matrícula d'honor:

- La matrícula d'honor es pot aconseguir amb una nota mitjana superior o igual a 9,0.
- Degut a que hi ha un nombre limitat de matrícules d'honor que es poden donar per grup, s'atorgaran per ordre de nota de major a menor.

Pràctiques, treballs o exàmens copiats:

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, i d'acord amb la normativa acadèmica vigent, les irregularitats comeses per un estudiant que puguin conduir a una variació de la qualificació es qualificaran amb un zero (0). 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. Aquestes irregularitats inclouen, entre d'altres:

- la còpia total o parcial d'una pràctica, informe, o qualsevol altra activitat d'avaluació;
- deixar copiar;
- presentar un treball de grup no fet íntegrament pels membres del grup;
- presentar com a propis materials elaborats per un tercer, encara que siguin traduccions o adaptacions, i en general treballs amb elements no originals i exclusius de l'estudiant;
- tenir dispositius de comunicació (com telèfons mòbils, smart watches, etc.) accessibles durant les proves d'avaluació teórico-pràctiques individuals (exàmens).

En cas que l'estudiant hagi comès irregularitats en un acte d'avaluació, la nota numèrica de l'expedient serà el valor menor entre 3.0 i la nota que li correspondria segons el mètode d'avaluació de l'assignatura (i per tant no serà possible l'aprovat per compensació).

En resum: copiar, deixar copiar o plagiar en qualsevol de les activitats d'avaluació equival a un SUSPENS amb nota inferior a 3.0.

Publicació notes, dates d'exàmens, etc:

Les dates d'avaluació continuada i lliurament de treballs es publicaran al campus virtual i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. Sempre s'informarà al campus virtual sobre aquests canvis ja que s'entén que és el mecanisme habitual d'intercanvi d'informació entre professor i estudiants.

Procediment de revisió de les qualificacions

Per a cada activitat d'avaluació, s'indicarà un lloc, data i hora de revisió en la que l'estudiant podrà revisar l'activitat amb el professor. En aquest context, 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.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Anàlisi semàntica de la pràctica	Veure activitats i instruments d'avaluació	1	0,04	1, 3, 4, 5, 6, 7
Anàlisi sintàctica de la pràctica	Veure activitats i instruments d'avaluació	1	0,04	1, 3, 4, 5, 6, 7
Examen de Pràctiques	Veure activitats i instruments d'avaluació	1	0,04	1, 2, 4
Examen final de teoria	Veure activitats i instruments d'avaluació	3	0,12	1, 2, 3, 4
Examen/s parcial/s	Veure activitats i instruments d'avaluació	2	0,08	1, 2, 3, 4
Generació de codi de la pràctica	Veure activitats i instruments d'avaluació	1	0,04	2, 4, 5, 6, 7

Bibliografia

Bibliografia bàsica

- *Charles N. Fischer & Richard J. Leblanc jr.*: Crafting a Compiler with C. The Benjamin / Cummings Publishing Company inc., 1991.
- *Jean-Paul Tremblay & Paul G. Sorenson*: The Theory and Practice of Compiler Writing. MCGRAW-HILL, 1985.
- *David A. Watt*: Programming Language Concepts and Paradigms. Prentice Hall, 1990.
- Alicia Garrido & all: Diseño de compiladores. Publicaciones Universidad de Alicante, 2002.

Bibliografia de consulta

- *David A. Watt*: Programming Language Processors. Prentice Hall, 1993.
- *Anthony J. Field & Peter G. Harrison*: Functional Programming. Addison-Wesley, 1988.
- *Terrence W. Pratt*: Programming Languages: Design and Implementation. Prentice Hall International Editions, 1984.
- *Edited by G. Goos and J. Hartmanis*: Compiler Construction. An Advanced Course. Springer-Verlag, 1974
- *Jean-Paul Tremblay & Paul G. Sorenson*: An Implementation Guide to Compiler Writing. MCGRAW-HILL, 1982.
- *Ronald Mak*: Writing Compilers & Interpreters: An Applied Approach. Wiley, 1991.
- *J. P. Brown*: Writing Interactive Compilers and Interpreters. Wiley, 1979.
- *Allen Y. Holub*: Compiler Design in C. Prentice Hall, 1990.