Compiladors 2013/2014

Codi: 102782
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OT	4	2

Professor de contacte

Nom: Francisco Javier Sánchez Pujadas
Correu electrònic: Javier.Sanchez.Pujadas@uab.cat

Utilització d´idiomes

Llengua vehicular majoritària: català (cat)
Algun grup íntegre en anglès: No
Algun grup íntegre en català: Sí
Algun grup íntegre en espanyol: No

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++ i Java facilitaran 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.
  - Reconeixedors 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.
- Treballar en equip

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 la capacitat d'anàlisi, síntesi i prospectiva.
5. Desenvolupar un mode de pensament i raonament crítics.
6. Treballar cooperativament.
7. Treballar de manera autònoma.

Continguts

Temari

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 lligam molt fort entre el que és la teoria, explicació de com funcionen i es poden crear compiladors, i la pràctica o els problemes, com crear un compilador realment. 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 lligats, donat que l'alumne dissenyarà una solució per resoldre un problema i la implementarà a la pràctica.

D'aquesta forma, el procés d'aprenentatge de l'alumne es fonamentarà 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 que els alumnes presentaran i discutiran les seves solucions

Classes de teoria

Les classes de teoria serviran per introduir els conceptes que es detallen al 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 ben 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 a 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 indicaran 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 a les pràctiques.

Pràctica

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

Activitats formatives

Títol	Hores	ECTS	Resultats d´aprenentatge
Tipus: Dirigides			
Classes de problemes	12	0,48	1, 2, 3, 4, 5
Classes de teoria	22	0,88	1, 2, 3, 4, 5
Tipus: Supervisades			
Sessions de pràctiques	12	0,48	1, 2, 3, 4, 6
Tipus: Autònomes			
Estudi de la matèria impartida en classes teòriques	36	1,44	1, 2, 3, 4, 5, 7
Preparació i implementació de la pràctica	47	1,88	1, 2, 3, 4, 5, 6
Preparació i resolució de problemes	12	0,48	1, 2, 3, 4, 5, 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 cada una de les parts de la pràctica.
  - **Nota individual**: És la puntuació obtinguda a les respostes individuals a preguntes durant les sessions presencials de pràctiques.
- **Nota Teoria**: correspon a l'examen final o exàmens parcials de teoria sobre l'assignatura. S'han d'aprovar per separat els dos exàmens parcials.

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

> **Nota Final** = (0.5 * **Nota Pràctiques**) + (0.5 * **Nota Teoria**)
>
> **Nota Pràctiques** = (0.5 * **Nota Individual**) + (0.5 * **Nota Grup**)
> **Nota Individual** = (0.2 * respostes A. Sintàctic) + (0.3 * respostes A. Semàntic) + (0.5 * respostes G. Codi)
> **Nota Grup** = (0.2 * Lliurament Sintàctic) + (0.3 * Lliurament semàntic) + (0.5 * Lliurament G. de Codi)

Condicions pel no presentat:

- No tenir cap part de l'assignatura suspesa

Condicions per aprovar l'assignatura:

- **Nota Final** >= 5
- **Nota pràctica** >= 5 i haver superar tots els autotests de la pràctica.
- **Nota teoria** >= 5 i exàmens parcials aprovats o les dues parts de l'examen final aprovades.

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). Per exemple, plagiar, copiar, deixar copiar, ..., una activitat d'avaluació, implicarà suspendre aquesta activitat d'avaluació 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.

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.

Activitats d´avaluació

Títol	Pes	Hores	ECTS	Resultats d´aprenentatge
Examen final de teoria (si no s'han aprovat els exàmens parcials)	100% nota de teoria	2	0,08	1, 2, 3, 4, 5
Examen parcial sobre anàlisis de codi font	50% nota de teoria	2	0,08	1, 3, 4, 5, 7
Examen parcial sobre generació de codi objecte	50% nota de teoria	2	0,08	2, 4, 5, 7
Lliurament i prova del bloc d'anàlisi semàntica de la pràctica	30% de la nota de pràctiques	1	0,04	1, 3, 4, 5, 6
Lliurament i prova del bloc d'anàlisi sintàctica de la pràctica	20% de la nota de pràctiques	1	0,04	1, 3, 4, 5, 6
Lliurament i prova del bloc de generació de codi de la pràctica	50% de la nota de pràctiques	1	0,04	2, 4, 5, 6

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.