

Compiladors I

Codi	Tipus	Curs/Semestre	Crèdits
20364	Troncal Semestral	2008 / 2009	6

Objectius

Competències específiques

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 genèriques

- Capacitat d'anàlisi i síntesi
- Treball en equip
- Motivació per la qualitat
- Raonament crític
- Aprenentatge autònom

Capacitats prèvies

Coneixements avançats de programació, estructura d'ordinadors i llenguatges formals.

Continguts

<p>1 Introducció. Conceptes bàsics.</p>	
<ol style="list-style-type: none"> 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. 	
<p>2. Anàlisi lexicogràfica: SCANNER.</p>	
<ol style="list-style-type: none"> 1. Definició d'anàlisi lexicogràfica 2. Expressions regulars 3. Implementació d'un analitzador lexicogràfic (Scanner) 4. Anàlisi lexicogràfica. 	
<p>3. Gramàtiques i anàlisi sintàctica: PARSER.</p>	
<ol style="list-style-type: none"> 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. 	
<p>4. Anàlisi semàntica.</p>	

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

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

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

6. Generació de codi.

1. Generació de codi.
2. Generació de codi d'expressions.
3. Procés de referències a estructures de dades.

7. Generalitats de l'aplicació de tècniques de compiladors per a la resolució de problemes.

1. Solució de problemes mitjançant tècniques de compiladors

Metodologia docent

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 es realitzarà en grups de 4-5 persones. La seva avaluació es farà al llarg del curs en sessions tutoritzades de 30 minuts. Cada grup d'alumnes tindrà assignades 6 sessions de 30 minuts de tutoria. El tutor encarregat de la sessió serà l'encarregat de fer el seguiment del treball. Les sessions de tutoria seran en un horari acordat pel professor i els alumnes dels presentats a continuació, l'horari serà el mateix per les 6 sessions.

Les sessions estaran dividides en 3 blocs: Anàlisi sintàctica, Anàlisi semàntica i Generació de codi. Per cada bloc hi haurà una primera sessió de presentació del disseny de la solució proposada, en la qual els alumnes explicaran la solució i el professor preguntarà sobre la mateixa i corregirà possibles errors. A la segona sessió els alumnes presentaran la solució ja implementada tot justificant-la i explicant el seu funcionament. En cada un dels tres blocs el professor proporcionarà abans de les entregues un joc de probes que ha de superar la solució implementada. Aquestes probes formaran el autotest que li servirà al alumne per poder comprovar el correcte funcionament de la solució ja implementada.

Per cada sessió hi haurà una nota individual resultat de les respostes donades al professor, i una nota col·lectiva resultant de la solució presentada

Avaluació

1a convocatòria (febrer/juny)		2a convocatòria (juliol/setembre)
Avaluació en grups	Avaluació individual	
Hi ha avaluació continuada de la part de pràctiques i problemes.	Hi ha avaluació continuada de la part de pràctiques i problemes.	
L'avaluació continuada consisteix en fer en grups, de com a màxim 5 persones, una pràctica que consta de tres fases a implementar: Anàlisi	L'avaluació continuada consisteix en fer en grups, de com a màxim 5 persones, una pràctica que consta de tres fases a implementar: Anàlisi	

<p>sintàctica, Anàlisi Semàntica i Generació de Codi. Cada una d'elles tindrà dues sessions d'avaluació, una de presentació del disseny i una d'implementació.</p> <p>L'avaluació del grup es fa de les sessions de lliurament de la implementació. La implementació ha de passar un joc de proves que els professors fan públic prèviament al lliurament. Si aquest joc de proves públic no es passa la pràctica està suspesa. A continuació els professors pasen un segon joc de proves per ajustar la nota entre el 5 i el 10.</p> <p>En primera convocatòria un alumne es considerarà no presentat només si compleix totes les condicions següents:</p> <ul style="list-style-type: none"> • No complir les condicions per aprovar. • No haver suspès l'À'examen de teoria. • No haver suspès la pràctica. 	<p>sintàctica, Anàlisi Semàntica i Generació de Codi. Cada una d'elles tindrà dues sessions d'avaluació, una de presentació del disseny i una d'implementació.</p> <p>L'avaluació de individual es fa fent preguntes de manera individual als estudiants per veure el seu nivell d'assoliment dels coneixements de l'assignatura. Si l'estudiant no sap respondre les preguntes correctament es considera que no te un nivell adequat per aprobar l'assignatura.</p> <p>A part hi ha un exàmen final de teoria tipu test.</p> <p>En primera convocatòria un alumne es considerarà no presentat només si compleix totes les condicions següents:</p> <ul style="list-style-type: none"> • No complir les condicions per aprovar. • No haver suspès l'À'examen de teoria. • No haver suspès la pràctica. 	<p>En segona convocatòria només es podrà recuperar la nota de l'À'examen de teoria.</p> <p>No es podrà tornar a fer l'À'entrega final de la pràctica. La nota pràctica s'À'obté per avaluació continuada que no es possible recuperar un cop ha acabat l'À'assignatura.</p> <p>En segona convocatòria un alumne es considerarà no presentat si compleix totes les condicions següents:</p> <ul style="list-style-type: none"> • No complir les condicions per aprovar. • No haver suspès l'À'examen de teoria de la segona convocatòria
---	---	---

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:* **Programing Language Concepts and Paradigms.** Prentice Hall, 1990.
- Alicia Garrido & all: **Diseño de compiladores.** Publicaciones Universidad de Alicante, 2002.

Bibliografia complementària

- *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.

Enllaços

[Pagina web de l'assignatura](http://www.cvc.uab.es/shared/teach/a20364/c20364.htm)

<http://www.cvc.uab.es/shared/teach/a20364/c20364.htm>