

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (*Asociación de Técnicos de Informática*). **Novática** edita también **Upgrade**, revista digital de **CEPIS** (*Council of European Professional Informatics Societies*), en lengua inglesa.

<<http://www.ati.es/novatica/>>
<<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (*Council of European Professional Informatics Societies*) y tiene un acuerdo de colaboración con **ACM** (*Association for Computing Machinery*). Tiene asimismo acuerdos de vinculación o colaboración con **AdaSpain**, **Ai2** y **ASTIC**.

CONSEJO EDITORIAL

Antoni Carbonell Noguera, Francisco López Crespo, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molas i Bertrán, Roberto Moya Quiles, César Pérez Chirinos, Mario Piattini Velasco, Fernando Pierra Gómez (Presidente del Consejo), Miquel Sarries Grifó, Asunción Yturbe Herranz

Coordinación Editorial

Rafael Fernández Calvo <rfcvalvo@ati.es>

Composición y autoedición

Jorge Liácer

Traducciones

Grupo de Lengua e Informática de ATI

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

SECCIONES TÉCNICAS: COORDINADORES

Administración Pública electrónica

Gumersindo García Arribas, Francisco López Crespo (MAP)

<gumersindo.garcia@map.es>, <lfc@ati.es>

Arquitecturas

Jordi Tubella (DAC-UPC) <jordi@ac.upc.es>

Victor Villos Yollera (Univ. de Zaragoza) <vic@unizar.es>

Auditoría SITIC

Marina Tourinho, Manuel Palao (ASIA)

<marinatourinho@marinatourinho.com>, <manuel@palao.com>

Bases de datos

Coral Calero Muñoz, Mario G. Piattini Velhuis

(Escuela Superior de Informática, UCLM)

<Coral.Calero@uclm.es>, <mpiattini@inf-cr.uclm.es>

Derecho y tecnologías

Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV) <ihernando@legaltek.net>

Isabel Davara Fernández de Marcos (Davara & Davara) <idadava@davara.com>

Enseñanza Universitaria de la Informática

Joaquín Ezpeleta Mateo (CPS-UZAR) <ezpeleta@posta.unizar.es>

Cristóbal Pareja Flores (DSIP-UCM) <cpareja@sip.ucm.es>

Informática y Filosofía

José Corco (UC) <jcorco@unica.edu>

Esperanza Marcos (ESCET-URJC) <cuca@escet.urjc.es>

Informática Gráfica

Roberto Vivo (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software

Javier Dolado Cosin (DLSI-UPV) <dolado@si.ehu.es>

Luis Fernández (PRIS-EL-UEM) <lufern@dpris.es.ueu.es>

Inteligencia Artificial

Federico Barber, Vicente Botti (DSIC-UPV)

<fbotti@dsic.upv.es>

Interacción Persona-Computador

Julio Abascal González (FI-UPV) <julio@si.ehu.es>

Jesús Lorés Vidal (Univ. de Lleida) <jesus@eup.udl.es>

Internet

Alonso Álvarez García (TID) <alonso@ati.es>

Llorenç Panes Casas (Indra) <pages@ati.es>

Lengua e Informática

M. del Carmen Ugarte (IBM) <cugarte@ati.es>

Lenguajes Informáticos

Andrés Marín López (Univ. Carlos III) <amarin@it.uc3m.es>

J. Ángel Velázquez (ESCET-URJC) <a.velazquez@escet.urjc.es>

Libertades e Informática

Alfonso Escalano (FIR-Univ. de La Laguna) <aescalano@ull.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>

Manuel Palomar (Univ. de Alicante) <mpalomar@dsi.ua.es>

Mundo estudiantil

Adolfo Vázquez Rodríguez

(Rama de Estudiantes del IEEE-UCM) <a.vazquez@ieee.org>

Profesión informática

Rafael Fernández Calvo (ATI) <rfcvalvo@ati.es>

Miquel Sarries Grifó (Ayto. de Barcelona) <msarries@ati.es>

Redes y servicios telemáticos

Luis Gujardo Coloma (DCOM-UPV) <lgujar@ddcom.upv.es>

Josep Solé Pareja (DAC-UPC) <pareja@ac.upc.es>

Seguridad

Javier Arellito (Redes y Sistemas, Bilbao) <jarellito@orion.deusto.es>

Xavier López Muñoz (ETSI Informática-UMA) <xljm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso, Juan Antonio de la Puente

(DI-UPM) <aalonso.jpuente@di.upm.es>

Software Libre

Jesús M. González Barahona, Pedro de las Heras Quirós

(GSYC-URJC) <[@gsyc.escet.urjc.es">jgb.pheras">@gsyc.escet.urjc.es](mailto:jgb.pheras)>

Tecnología de Objetos

Jesús García Molina (DIS-UM) <jmolina@correo.um.es>

Gustavo Rossi (LIFIA-UNLP, Argentina) <gustavo@sol.info.unpl.edu.ar>

Tecnologías para la Educación

Josep Sales Rull (ESPRAL) <jsales@pie.xtec.es>

Tecnologías y Empresa

Pablo Hernández Medrano (Bluematt) <pablohm@bluematt.biz>

TIC para la Sanidad

Valentín Masero Vargas (DI-UNEX) <vmasero@unex.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)

<[@lcc.uma.es">aguayo.guevara">@lcc.uma.es](mailto:aguayo.guevara)>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos.

Novática permite la reproducción de todos los artículos, salvo los marcados con © o *copyright*, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial y Redacción Central (ATI Madrid)

Padilla 66, 3º, dcha., 28006 Madrid

Tfno. 91 4029391; fax 91 3093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Reino de Valencia 23, 46005 Valencia

Tfno./fax 96 3330392 <secreval@ati.es>

Administración y Redacción ATI Cataluña

Via Laietana 41, 1º, 1ª, 08003 Barcelona

Tfno. 93 4125235; fax 93 4127713 <secregen@ati.es>

Redacción ATI Andalucía

Isaac Newton, s/n, Ed. Sadlet

Isla Cartuja 41092 Sevilla. Tfno./fax 95 4460779 <secreand@ati.es>

Redacción ATI Aragón

Lagasca 9, 3-B, 50006 Zaragoza

Tfno./fax 97 6251181 <secreara@ati.es>

Redacción ATI Asturias-Cantabria

<ag-astucant@ati.es>

Redacción ATI Castilla-La Mancha

<ag-clmancha@ati.es>

Redacción ATI Galicia

Recinto Ferial s/n, 36540 Silleda (Pontevedra)

Tfno. 98 6581413; fax 98 6580162 <secregal@ati.es>

Suscripción y Ventas

<<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid

Publicidad

Padilla 66, 3º, dcha., 28006 Madrid

Tfno. 91 4029391; fax 91 3093685 <novatica.publicidad@ati.es>

Imprenta

9 Impresión S.A. Juan de Austria 66, 08005 Barcelona

Depósito legal: B 15.154-1975 - ISSN: 0211-2124; CODEN NOVAEC

Portada: Antonio Crespo Foix / © ATI 2003

Diseño: Fernando Agresta / © ATI 2003

Nº 165, septiembre-octubre 2003
Edición digital

sumario

editorial

ATI ante las patentes de software

> 02

en resumen

Un nuevo diseño

> 02

Rafael Fernández Calvo

monografía

Teleaprendizaje / e-Learning

(En colaboración con *Upgrade*)

Editores invitados: *Ángel Fidalgo Blanco* y *Martín Llamas Nistal*

Presentación. Aprendiendo a distancia

> 03

Ángel Fidalgo Blanco, Martín Llamas Nistal

Aprendizaje potenciado por la tecnología: actividades de investigación en el marco de la Comisión Europea

> 06

Patricia Manson, Elena Coello

Estandarización en aprendizaje basado en ordenador

> 08

Judith Rodríguez Estévez, Manuel Caeiro Rodríguez, Juan M. Santos Gago

FEI organismo europeo de estandarización para las Tecnologías del Aprendizaje

> 15

Frans Van Assche, Mike Collett

Ingeniería de Software basada en componentes y CSCL

en el campo del teleaprendizaje

> 19

Yannis A. Dimitriadis, Juan Ignacio Asensio Pérez,

Alejandra Martínez Monés, César A. Osuna Gómez

AVANTE: una arquitectura para instrucción basada en la web sobre estándares XML/XSL, software libre y componentes CORBA

> 26

Victor Theokisto, Adelaide Bianchini, Edna Ruckhaus, Lee Lima

e-Learning en la formación a distancia y en los nuevos contextos corporativos

> 34

Enrique Rubio Royo, Domingo J. Gallego, Catalina Alonso García

Las tecnologías y la gestión del conocimiento

para la formación continua del médico colegiado

> 40

Cristina Zamanillo Sarmiento, Julián Ruiz Ferrán, Ángel Fidalgo Blanco

EducaNext: un servicio para la compartición de conocimiento

> 45

Joaquín Salvachúa Rodríguez, Juan Quemada Vives, Blanca Rodríguez Pajares,

Gabriel Huecas Fernández-Toribio

secciones técnicas

administración pública electrónica

e-Administración y "Sociedad en Red"

> 51

Narciso Pizarro y Ponce de la Torre

informática y filosofía

Inteligencia, memoria social y ciberespacio

> 57

Isidro Ramos, Jennifer Pérez

tecnología de objetos

Metodos formales orientados a objetos

> 62

Francisco José Galán Morillo, José Miguel Cañete Valdeón

Referencias autorizadas

> 65

sociedad de la información

if

Waternet: una revolución tecnológica en la Red

> 72

Tadeo Fernández Trullenque

personal y transferible

Locos por los ordenadores (IV): "La prodigiosa mente de John Von Neumann"

> 73

Rafael Fernández Calvo

programar es crear

CUPCAM 2003: I Concurso Universitario de Programación de la Comunidad Autónoma de Madrid

> 75

Ángel Herranz Nieva, Cristóbal Pareja Flores

¿Dónde está mi interrupción? - (CUPCAM 2003, problema A)

> 76

Manuel Carro, Óscar Martín

asuntos internos

Normas de publicación para autores / Socios Institucionales

> 77

Monografía del próximo número: "Planes de Contingencia y Seguridad TIC"

Manuel Carro¹, Óscar Martín²

¹ Facultad de Informática, Universidad Politécnica de Madrid; ² Unidad de Organización y Sistemas, Caja Madrid

<mcarro@fi.upm.es>, <oscar_m@personas.net>

¿Dónde está mi interrupción?

I Concurso Universitario de Programación de la Comunidad Autónoma de Madrid (CUPCAM 2003): enunciado del problema A

Los microprocesadores modernos pueden interrumpir el flujo de control de un programa cuando se solicita, bien desde *hardware* (respondiendo al funcionamiento de un dispositivo físico), bien desde *software*. El control puede saltar entonces a una rutina de atención previamente especificada (el *manejador de interrupciones*), atender la interrupción ejecutando el código de la rutina y continuar con el programa en el punto en que su flujo fue derivado. Asumiremos que las interrupciones están numeradas desde

```
Manejador ::= Instrucción; Manejador | Instrucción;
Instrucción ::= Registro := Op BinOp Op
BinOp      ::= and | or | xor
Op         ::= Valor | not Valor
Valor      ::= Registro | Número
Registro   ::= imri
Número     ::= 0 | 1 | 2 | ... | 2w-1
```

Figura 1. Lenguaje de manejo de interrupciones.

1 hasta w , $1 \leq w \leq 32$, y que cada interrupción está asociada a un manejador distinto.

La recepción de la interrupción i puede ser habilitada o deshabilitada ajustando el bit i -ésimo de un registro de w bits

```
imr1 := not imr0 or not 23;
imr2 := imr1 xor not imr0;
imr3 := 87 or imr2; imr0 :=
not imr3 and imr1;
```

Figura 2. Un programa que puede deshabilitar todas las interrupciones.

(denominado *máscara de interrupciones*) a 1 (desbloqueo) o a 0 (bloqueo). En aras de la simplicidad, identificaremos de ahora en adelante la máscara de interrupciones con una variable denominada *imr0*. Simplificando, cada manejador de interrupciones puede cambiar el valor de

imr0 para deshabilitar la recepción de la interrupción que está siendo atendida (y así evitar caer en una sucesión ilimitada de reentradas en la rutina) y también para deshabilitar la recepción de otras interrupciones que tienen menos prioridad.¹

```
imr1 := imr0 or not 23; imr2
:= imr1 xor not imr0; imr3
:= 87 or imr2; imr0 := not
imr3 or imr1;
```

Figura 3. Un programa que nunca deshabilitará todas las interrupciones.

Conocer el valor que un manejador de interrupciones deja en *imr0* es extremadamente interesante: si su valor final es 0, entonces ninguna otra interrupción podrá ser recibida tras el final de ese manejador. Nos interesa detectar si esta situación es posible analizando los valores que los manejadores de interrupciones pueden dejar en *imr0*. Las operaciones para manejar el valor de la máscara de interrupciones *imr0* están expresadas por el lenguaje que aparece en la **figura 1**, donde *imr*i**, $i \neq 0$, son variables de w bits. Asumiremos, por tanto, que no hay saltos condicionales ni bucles.

En las **figuras 2** y **3** tenemos dos ejemplos de programas que asumen $w = 8$. El programa de la **figura 2** deja un valor final 0 en *imr0* para los valores iniciales de *imr0* pertenecientes a $\{168...175\} \cup \{184...191\} \cup \{232...239\} \cup \{248...255\}$, mientras que el de la **figura 3** nunca dejará un valor *imr0* = 0.

La tarea a realizar es leer una serie de manejadores de interrupciones y determinar si es posible que, para algún valor inicial de *imr0*, el valor final de *imr0* sea 0, o si para *ningún* valor inicial de *imr0* su valor final será 0.

Nota

¹ En una arquitectura real existirían también interrupciones que no se pueden deshabilitar, cuya existencia ignoraremos aquí.

Descripción de la entrada

La entrada consistirá en el código de una serie de manejadores de interrupciones. Cada definición empieza con un número natural k , $0 < k \leq 20$, que se corresponde con el número de instrucciones de que consta el código del manejador y un número natural w que indica el número de interrupciones (... número de bits en cada *imr*i**) a considerar. A continuación siguen k líneas con una instrucción en cada una de ellas. En un manejador con k instrucciones, *imr*i** puede variar de *imr0* a *imr*k**. Con la excepción de *imr0*, ningún *imr*i** se usará en la parte derecha de una asignación antes de haber aparecido previamente en la parte izquierda de otra asignación.

Descripción de la salida

Para cada manejador, el programa debe imprimir la palabra **cero** si es posible que el manejador deje un valor 0 en *imr0* y no **cero** si esta situación es imposible para cualquier valor inicial de *imr0*.

Ejemplo de entrada

```
4 8
imr1 := not imr0 or not 23;
imr2 := imr1 xor not imr0; imr3 := 87 or imr2; imr0 := not imr3 and
imr1;
4 7
imr1 := imr0 or not 23;
imr2 := imr1 xor not imr0; imr3 := 87 or imr2; imr0 := not
imr3 or imr1;
1 32
imr0 := imr0 and imr0;
1 32
imr0 := imr0 or not imr0;
```

Salida para el ejemplo de entrada

```
cero
no cero
cero
no cero
```