structura de Computadores

Programación a Nivel-Máquina I: Conceptos Básicos

Estructura de Computadores Semana 2

Bibliografía:

[BRY11] Cap.3

Computer Systems: A Programmer's Perspective. Bryant, O'Hallaron. Pearson, 2011

Signatura ESIIT/C.1 BRY com

Transparencias del libro CS:APP, Cap.3

Introduction to Computer Systems: a Programmer's Perspective **Autores:** Randal E. Bryant y David R. O'Hallaron

1

Grado Informática, 2º Curso

Estructura de Computadores

Guía de trabajo autónomo (4h/s)

- Lectura: del Cap.3 CS:APP (Bryant/O'Hallaron)
 - Historical perspective, Program Encodings, Data Formats, Accessing Info.
 - 3.1 3.4 pp.187-211
 - Procedures, Stack frame, Transferring control (opcional por ahora)
 - 3.7.1 3.7.2 pp.253-257
 - x86-64, History, Overview, Accessing Info., hasta Arithmetic
 - 3.13 3.13.3 pp.301-311
- **Ejercicios:** del Cap.3 CS:APP (Bryant/O'Hallaron)
 - Probl. 3.1 3.5pp.204, 208, 210-211
 - Probl. 3.30 p.257
 - Probl. 3.46 3.47 pp.305, 310

Bibliografía:

[BRY11] Cap.3

Computer Systems: A Programmer's Perspective. Bryant, O'Hallaron. Pearson, 2011

Signatura ESIIT/C.1 BRY com

_

structura de Computadores

Programación Máquina I: Conceptos Básicos

- Historia de los procesadores y arquitecturas de Intel
- Lenguaje C, ensamblador, código máquina
- Conceptos básicos asm: Registros, operandos, move
- Intro a x86-64

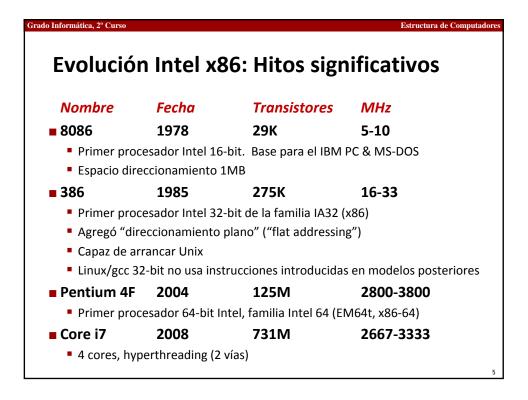
3

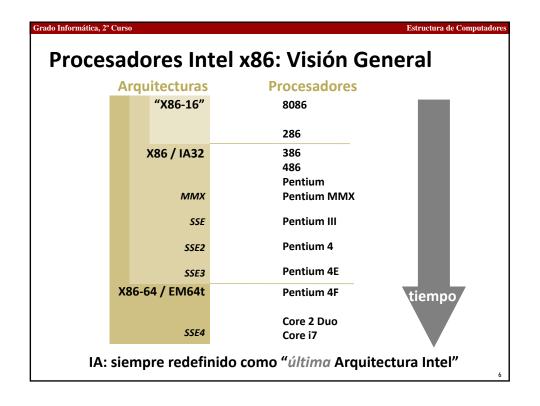
Grado Informática, 2º Curso

Estructura de Computadores

Procesadores Intel x86

- Dominan totalmente mercado portátil/sobremesa/servidor
- Diseño evolutivo
 - Compatible ascendentemente hasta el 8086, introducido en 1978
 - Va añadiendo características conforme pasa el tiempo
- Computador con repertorio instrucciones complejo (CISC)
 - Muchas instrucciones diferentes, con muchos formatos distintos
 - Pero sólo un pequeño subconjunto aparece en programas Linux
 - Difícil igualar prestaciones Computadores Repertorio Instr. Reducido (RISC)
 - Sin embargo, Intel lo ha conseguido
 - En lo que a velocidad se refiere. No tanto en (bajo) consumo.





Procesadores Intel x86, cont. ■ Evolución de las máquinas Integrated Memory Controller - 3 Ch DDR3 **386** 1985 0.3M Pentium 1993 3.1M Pentium/MMX 1997 4.5M PentiumPro 1995 6.5M Pentium III 1999 8.2M Pentium 4 2001 42M Shared L3 Cache Core 2 Duo 291M 2006 2008 731M Core i7

■ Características añadidas

- Instrucciones de soporte para operación multimedia
 - Operaciones paralelas con datos 1B, 2B y 4-byte, tanto enteros & p. flot.

Microfotografía de un dado Core i7

Instrucciones para posibilitar operaciones condicionales más eficientes

■ Evolución Linux/GCC

2 etapas principales: 1) soporte 32-bit 386. 2) soporte 64-bit x86-64

Más Información

Procesadores Intel (Wikipedia)

Microarquitecturas Intel

List_of_Intel_microprocessors

ark.intel.com

List_of_Intel_CPU_microarchitectures

Wikipedia EMT64 / EM64T / x86_64

structura de Computadores

Nueva especie: ia64, IPF*, Itanium,...

Nombre Fecha Transistores

■ Itanium 2001 10M

- Primer intento de arquitectura 64-bit: inicialmente denominada IA64
- Repertorio instr. radicalmente nuevo, diseñado para altas prestaciones
- Puede ejecutar programas IA32 anteriores
 - "motor x86 a bordo"
- Proyecto conjunto con Hewlett-Packard

■ Itanium 2 2002 221M

Gran aumento de prestaciones

■ Itanium 2 Dual-Core 2006 1.7B

■ Itanium no ha cuajado en el mercado

- Sin compatibilidad ascendente, sin apoyo buen compilador
- Pentium 4 salió "demasiado bueno"

* "IPF" = Itanium Processor Family 9

Grado Informática, 2º Curso

Estructura de Computadores

Clones x86: Advanced Micro Devices (AMD)

■ Históricamente

- AMD ha ido siguiendo a Intel en todo
- CPUs un poco más lentas, mucho más baratas

Y entonces

- Reclutaron los mejores diseñadores de circuitos de Digital Equipment Corp. y otras compañías con tendencia descendente
- Construyeron el Opteron: duro competidor para el Pentium 4
- Desarrollaron x86-64, su propia extensión a 64 bits

Estructura de Computadore

Los 64-Bit de Intel

- Intel intentó un cambio radical de IA32 a IA64
 - Arquitectura totalmente diferente (Itanium)
 - Ejecuta código IA32 sólo como herencia*
 - Prestaciones decepcionantes
- AMD intervino con una Solución Evolutiva
 - x86-64 (ahora llamado "AMD64")
- Intel se sintió obligado a concentrarse en IA64
 - Difícil admitir error, o admitir que AMD es mejor
- 2004: Intel anuncia extensión EM64T de la IA32
 - Extended Memory 64-bit Technology
 - Casi idéntica a x86-64
- Todos los procesadores x86 salvo gama baja soportan x86-64
 - Pero gran cantidad de código se ejecuta aún en modo 32-bits

* "legacy" = herencia de características 11

Grado Informática, 2º Curso

Estructura de Computadores

Nosotros veremos:

- IA32
 - El x86 tradicional
- x86-64/EM64T
 - El "nuevo" estándar
 - Comentaremos brevemente las diferencias
- Presentación
 - El libro presenta IA32 en las Secciones 3.1—3.12
 - Cubre x86-64 en 3.13
 - Nosotros los veremos a la vez
 - Las prácticas se concentran en IA32

structura de Computadores

Programación Máquina I: Conceptos Básicos

- Historia de los procesadores y arquitecturas de Intel
- Lenguaje C, ensamblador, código máquina
- Conceptos básicos asm: Registros, operandos, move
- Intro a x86-64

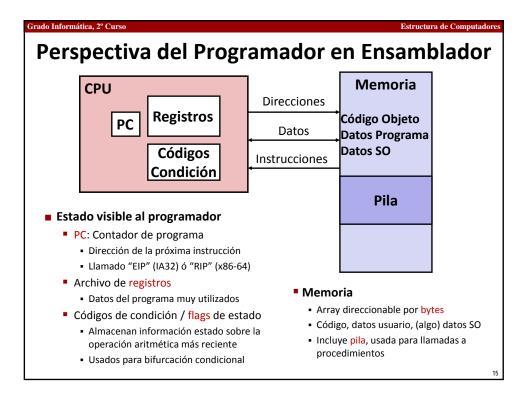
13

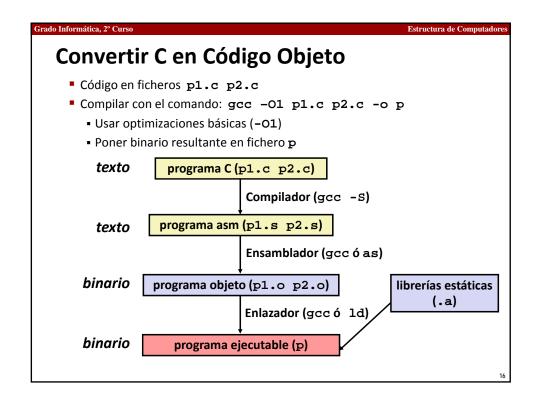
Grado Informática, 2º Curso

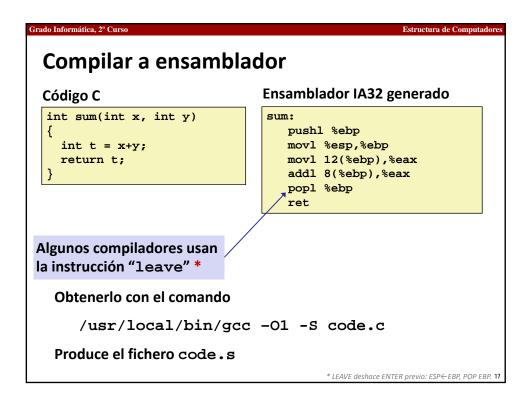
Estructura de Computadores

Definiciones

- Arquitectura: (también arquitectura del repertorio de instrucciones: ISA) Las partes del diseño de un procesador que se necesitan entender para escribir código ensamblador.
 - Ejemplos: especificación del repertorio de instrucciones, registros.
- Microarquitectura: Implementación de la arquitectura.
 - Ejemplos: tamaño de las caches y frecuencia de los cores.
- Ejemplo de ISAs (Intel): x86, IA, IPF







Grado Informática, 2º Curso Estructura de Computadores

Características Ensamblador: Tipos de Datos

- Datos "enteros" de 1, 2, ó 4 bytes
 - Valores de datos
 - Direcciones (punteros sin tipo)
- Datos en punto flotante de 4, 8, ó 10 bytes
- No hay tipos compuestos como arrays o estructuras
 - Tan sólo bytes ubicados contiguamente (uno tras otro) en memoria

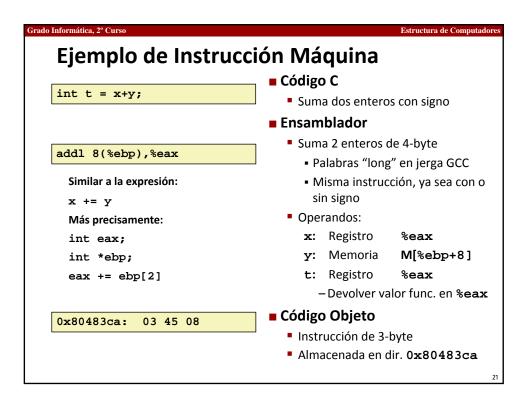
structura de Computadores

Características ensamblador: Operaciones

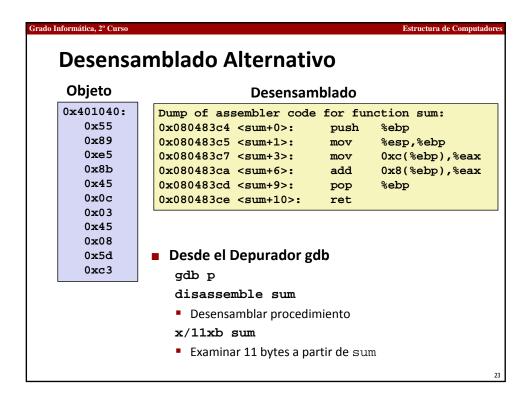
- Realizan función aritmética sobre datos en registros o memoria
- Transfieren datos entre memoria y registros
 - Cargar datos de memoria a un registro
 - Almacenar datos de un registro en memoria
- Transferencia de control
 - Llamadas incondicionales a procedimientos / retornos desde ellos
 - Saltos condicionales

19

Grado Informática, 2º Curso Estructura de Computadore Código Objeto Código para sum Ensamblador 0x401040 <sum>: ■ Traduce .s pasándolo a .o 0x55 Instrucciones codificadas en binario 0x890xe5 Imagen casi completa del código ejecutable d8x0 Le faltan enlaces entre código de ficheros 0x45diferentes 0x0c0x03Enlazador 0x45• Resuelve referencias entre ficheros 80x0• 11 bytes total Combina con libs. de tiempo ejec. estáticas* 0x5d• Cada instrucción 0xc3P.ej., código para malloc(), printf() 1, 2, ó 3 bytes Algunas libs. son dinámicamente enlazadas** • Empieza en direcc. • En enlace ocurre cuando el programa 0x401040 empieza a ejecutarse ** "dynamically linked libraries", o también "shared libs" * "static run-time libraries" = bibliotecas estáticas para soporte en tiempo de ejecución 20



Estructura de Computadore **Desensamblando Código Objeto** Desensamblado 080483c4 <sum>: 80483c4: 55 push %ebp 80483c5: 89 e5 mov 80483c7: 8b 45 0c mov 80483ca: 03 45 08 add mov %esp,%ebp 0xc(%ebp),%eax 0x8(%ebp),%eax 80483cd: 5d 80483ce: c3 ret Desensamblador objdump -d p Herramienta útil para examinar código objeto Analiza el patrón de bits de series de instrucciones Produce versión aproximada del código ensamblador (correspondiente) Puede ejecutarse sobre el fich. a . out (ejecutable completo) ó el . o

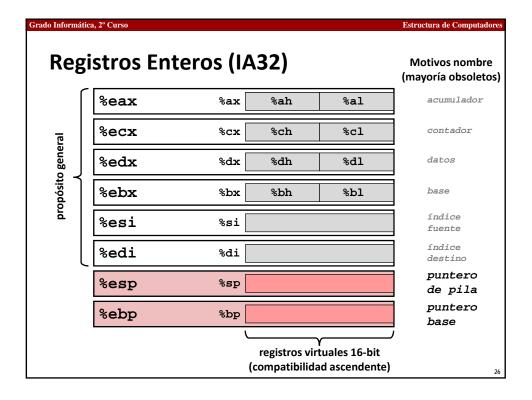


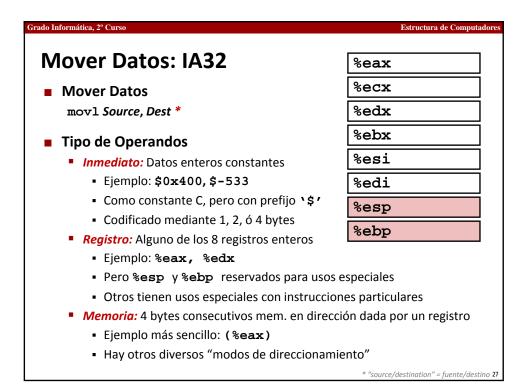
```
Grado Informática, 2º Curso
                                                     Estructura de Computadore
   ¿Qué se puede Desensamblar?
   % objdump -d WINWORD.EXE
   WINWORD.EXE:
                 file format pei-i386
   No symbols in "WINWORD.EXE".
   Disassembly of section .text:
   30001000 <.text>:
   30001000: 55
                             push
                                     %ebp
                      mov
   30001001: 8b ec
                                     %esp,%ebp
   30001003: 6a ff push
30001005: 68 90 10 00 30 push
                                     $0xffffffff
                                     $0x30001090
   3000100a: 68 91 dc 4c 30 push
                                     $0x304cdc91

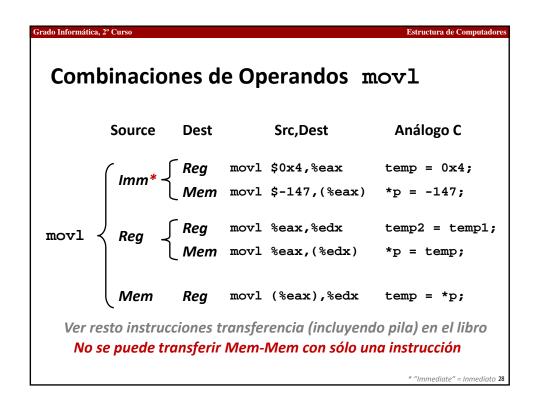
    Cualquier cosa que se pueda interpretar como código ejecutable

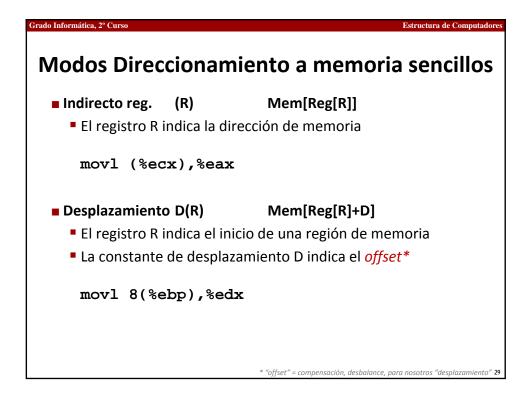
 El desensamblador examina bytes y reconstruye el fuente asm.
```

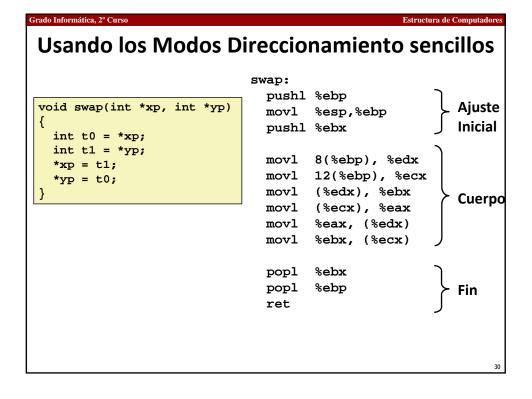
Programación Máquina I: Conceptos Básicos Historia de los procesadores y arquitecturas de Intel Lenguaje C, ensamblador, código máquina Conceptos básicos asm: Registros, operandos, move Intro a x86-64

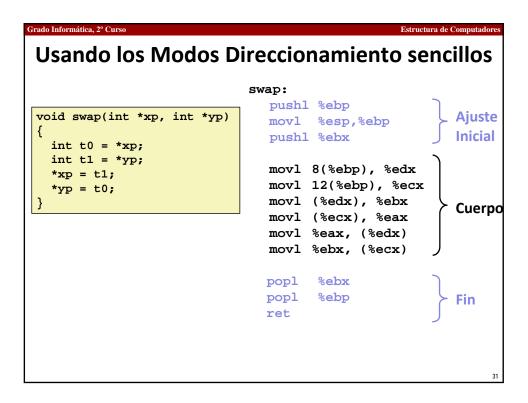


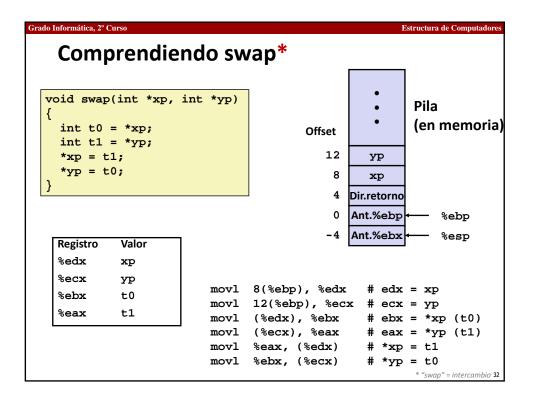


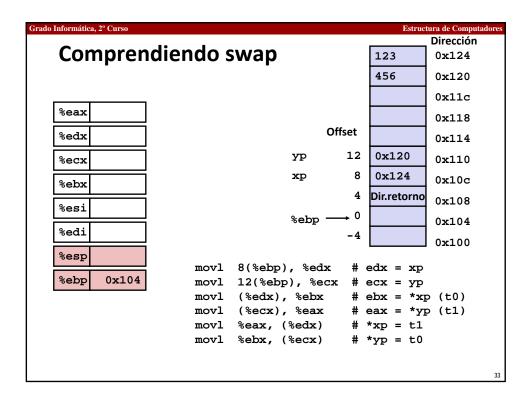


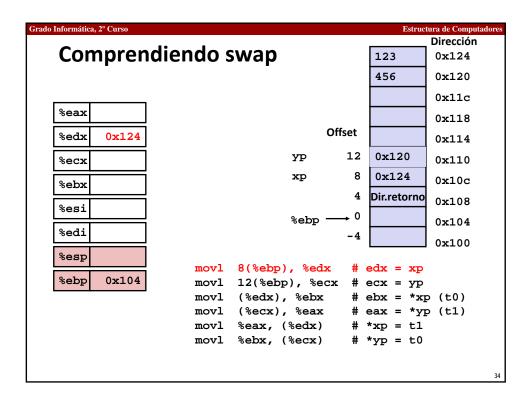


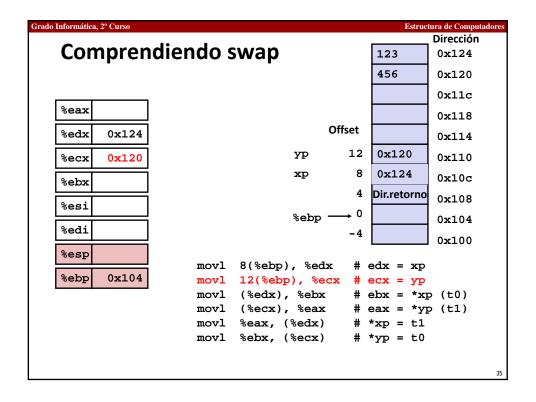


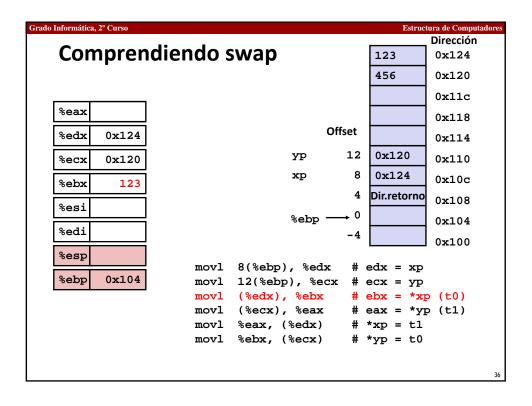


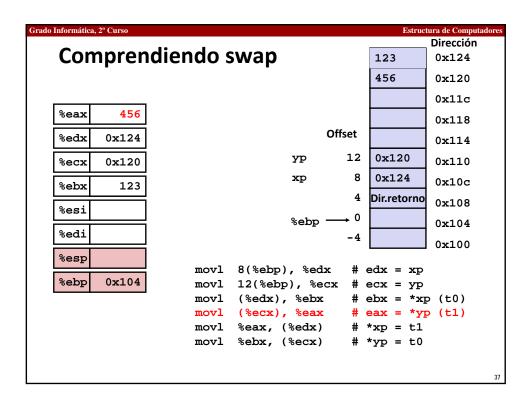


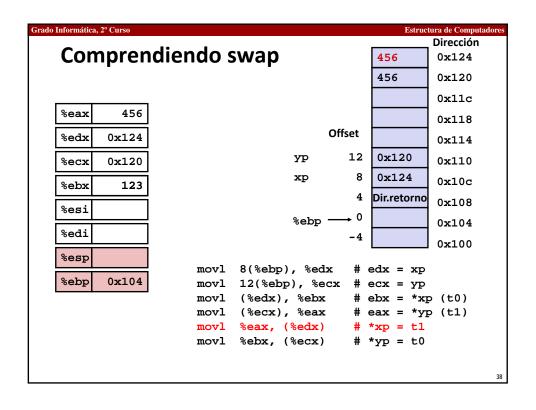


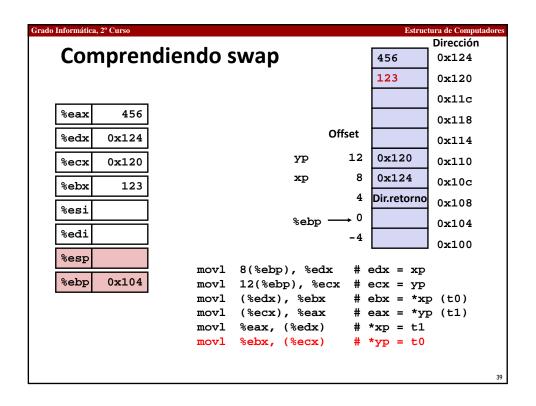


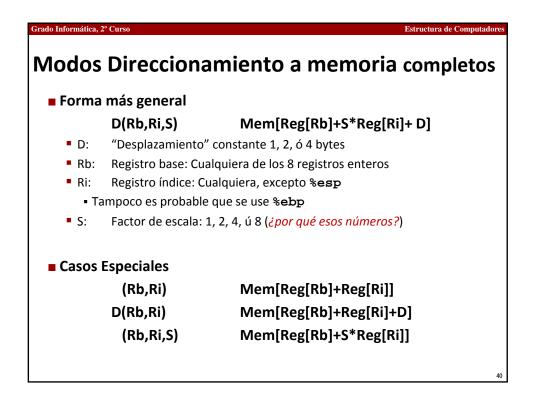












Programación Máquina I: Conceptos Básicos Historia de los procesadores y arquitecturas de Intel Lenguaje C, ensamblador, código máquina Conceptos básicos asm: Registros, operandos, move Intro a x86-64

Tamaño de Objetos C (en Bytes)			
Tipo de Datos C	Normal 32-bit	Intel IA32	x86-64
unsigned	4	4	4
• int	4	4	4
long int	4	4	8
char	1	1	1
short	2	2	2
float	4	4	4
double	8	8	8
 long double 	8	10/12	16
• char *	4	4	8



Instrucciones

palabra Long I (4 Bytes) ↔ palabra Quad q (8 Bytes)

Nuevas instrucciones:

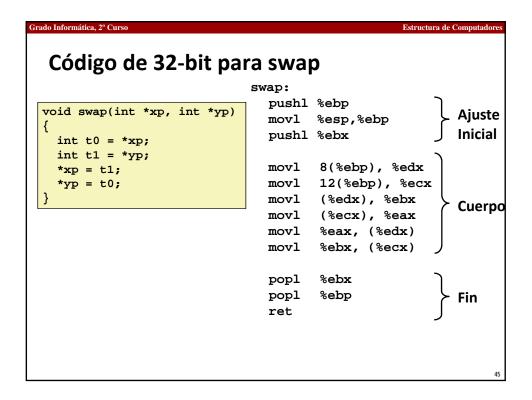
movl → movq

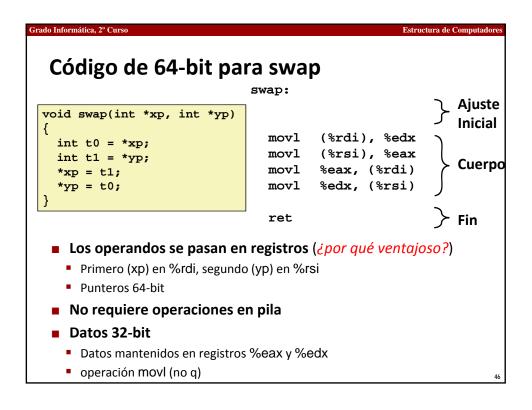
addl → addq
sall → salq
etc.

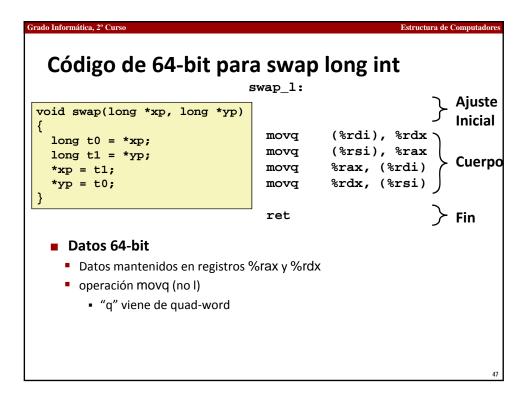
instrucciones 32-bit que generan resultados 32-bit

Ajustan los bits más significativos del registro destino a 0

Ejemplo: addl







Programación Máquina I: Resumen

.

- Historia de los procesadores y arquitecturas de Intel
 - Diseño evolutivo lleva a muchas peculiaridades y artefactos
- Lenguaje C, ensamblador, código máquina
 - El compilador debe transformar sentencias, expresiones, procedimientos, en secuencias de instrucciones de bajo nivel
- Conceptos básicos asm: Registros, operandos, move
 - Las instrucciones move x86 cubren amplia gama de formas de movimiento de datos (transferencia)
- Intro a x86-64
 - Un cambio importante respecto al estilo de código visto en IA32

Guía de trabajo autónomo (4h/s) ■ Estudio: del Cap.3 CS:APP (Bryant/O'Hallaron) Historical perspective, Program Encodings, Data Formats, Accessing Info. - 3.1 - 3.4 pp.187-211 • Probl. 3.1 - 3.5 pp.204, 208, 210-211 Procedures, Stack frame, Transferring control (opcional por ahora) - 3.7.1 - 3.7.2 pp.253-257 Probl. 3.30 p.257 x86-64, History, Overview, Accessing Info., hasta Arithmetic - 3.13 - 3.13.3 pp.301-311 Probl. 3.46 - 3.47 pp.305, 310 Bibliografía: [BRY11] Cap.3 Computer Systems: A Programmer's Perspective. Bryant, O'Hallaron. Pearson, 2011 Signatura ESIIT/C.1 BRY com