1.

APA ITU BAHASA CAMPURAN

Dalam pemrograman komputer , bahasa rakitan (atau bahasa assembler atau kode mesin simbolis ), sering disebut hanya sebagai rakitan dan biasa disingkat ASM atau asm adalah bahasa pemrograman tingkat rendah dengan korespondensi yang sangat kuat antara instruksi dalam bahasa dan instruksi kode mesin arsitektur. Bahasa rakitan biasanya memiliki satu pernyataan perinstruksi mesin tetapi konstanta, komentar,arahan assembler,label simbolik,misalnya lokasi memori , register,dan makro umumnya juga didukung.

Terkadang ada lebih dari satu assembler untuk arsitektur yang sama, dan terkadang assembler dikhususkan untuk suatu sistem operasi atau sistem operasi tertentu. Sebagian besar bahasa rakitan tidak menyediakan sintaks khusus untuk panggilan sistem operasi, dan sebagian besar bahasa rakitan dapat digunakan secara universal dengan sistem operasi apa pun, karena bahasa tersebut menyediakan akses ke semua kemampuan nyata dari prosesor , yang menjadi dasar semua mekanisme panggilan sistem akhirnya istirahat. Berbeda dengan bahasa assembly, sebagian besar bahasa pemrograman tingkat tinggi umumnya portabel di berbagai arsitektur tetapi memerlukan interpretasi atau kompilasi , tugas yang jauh lebih rumit daripada perakitan.

BAHASA RAKITAN

Bahasa rakitan menggunakan mnemonik untuk mewakili, misalnya, setiap instruksi mesin tingkat rendah atau opcode , setiap arahan , biasanya juga setiap register arsitektur , flag , dll. Beberapa mnemonik mungkin sudah ada di dalamnya dan beberapa ditentukan oleh pengguna. Banyak operasi memerlukan satu atau lebih operan untuk membentuk instruksi yang lengkap. Kebanyakan assembler mengizinkan konstanta, register, dan label bernama untuk lokasi program dan memori, dan dapat menghitung ekspresi untuk operan. Dengan demikian, programmer terbebas dari perhitungan berulang yang membosankan dan program assembler jauh lebih mudah dibaca daripada kode mesin. sangat umum untuk mesin yang menggunakan kartu berlubang pada tahun 1950an dan awal 1960an. Beberapa assembler memiliki sintaks bentuk bebas, dengan bidang dipisahkan oleh pembatas, misalnya tanda baca, spasi . Beberapa assembler bersifat hibrid, dengan, misalnya, label, pada kolom tertentu dan bidang lain yang dipisahkan oleh pembatas; ini menjadi lebih umum daripada sintaksis berorientasi kolom pada tahun 1960an.

TERMINOLOGI

Assembler makro adalah assembler yang menyertakan fasilitas instruksi makro sehingga teks bahasa assembly (yang diparameterisasi) dapat diwakili oleh sebuah nama, dan nama tersebut dapat digunakan untuk menyisipkan teks yang diperluas ke dalam kode lain.

Cross assembler (lihat juga cross compiler ) adalah assembler yang dijalankan pada komputer atau sistem operasi ( sistem host ) dengan jenis yang berbeda dari sistem tempat kode yang dihasilkan akan dijalankan ( sistem target ).

Assembler tingkat tinggi adalah program yang menyediakan abstraksi bahasa yang lebih sering dikaitkan dengan bahasa tingkat tinggi, seperti struktur kontrol tingkat lanjut

Microassembler adalah program yang membantu menyiapkan program mikro , yang disebut firmware , untuk mengontrol pengoperasian komputer tingkat rendah.

Contoh Program Bahasa Campura

1. Code-Switching dalam Media Sosial:

Di era digital, seringkali kita menemukan orang menggunakan bahasa campura dalam media sosial. Misalnya, di antara kalimat berbahasa Indonesia, mereka mungkin menyisipkan kata-kata berbahasa Inggris atau bahasa daerah.

Contoh: "Aku seneng banget dengan weekend ini, gonna hang out sama teman-teman di café yang cozy."

2. Iklan Multibahasa:

Banyak perusahaan menggunakan bahasa campura dalam iklan mereka untuk menjangkau berbagai kelompok masyarakat. Iklan tersebut bisa memadukan bahasa lokal dengan bahasa nasional atau bahkan bahasa internasional.

Contoh: "Diskon gede-gedean! Grab it now sebelum kehabisan!"

3. Percakapan Sehari-hari:

Dalam percakapan sehari-hari, orang seringkali menggunakan bahasa campura untuk menggambarkan nuansa perasaan, humor, atau hanya untuk memperkaya komunikasi.

Contoh: "Nggak ada masalah, bro. Kita bisa settle ini dengan meeting besok di kantor. No worries, ya!"

2.

A. Syntax untuk Deklarasi Data Dasar (String, Integer, Boolean, Float 32-bit):

String:

assembly

Copy code

section .data

myString db 'Hello, World!', 0

Dalam contoh ini, myString adalah label yang menunjukkan string "Hello, World!" dan db menunjukkan byte data.

Integer:

assembly

Copy code

section .data

myInteger dd 42

Deklarasi integer menggunakan dd untuk menunjukkan data double-word (32-bit).

Boolean:

assembly

Copy code

section .data

myBoolean db 1

Boolean sering diwakili oleh byte, di sini myBoolean memiliki nilai 1 untuk true.

Float 32-bit:

assembly

Copy code

section .data

myFloat dd 3.14

Deklarasi float menggunakan dd untuk data double-word (32-bit).

B. Syntax untuk Branching (if ... else ...) Beserta Variasinya:

Jump if Equal (JE):

assembly

Copy code

section .text

global \_start

\_start:

mov eax, 5

cmp eax, 5

je equal\_branch ; jika sama, lompat ke equal\_branch

jmp not\_equal\_branch ; jika tidak sama, lompat ke not\_equal\_branch

equal\_branch:

; kode untuk kasus jika sama

jmp end\_program

not\_equal\_branch:

; kode untuk kasus jika tidak sama

end\_program:

; kode program berakhir

Dalam contoh ini, je digunakan untuk lompat jika equal (sama).

Jump if Not Equal (JNE):

assembly

Copy code

section .text

global \_start

\_start:

mov eax, 5

cmp eax, 5

jne not\_equal\_branch ; jika tidak sama, lompat ke not\_equal\_branch

jmp equal\_branch ; jika sama, lompat ke equal\_branch

equal\_branch:

; kode untuk kasus jika sama

jmp end\_program

not\_equal\_branch:

; kode untuk kasus jika tidak sama

end\_program:

; kode program berakhir

Dalam contoh ini, jne digunakan untuk lompat jika tidak sama.

Penting untuk dicatat bahwa instruksi dan label dalam bahasa assembly dapat bervariasi tergantung pada assembler yang digunakan dan sintaksis tertentu (misalnya, AT&T atau Intel syntax). Contoh di atas menggunakan sintaks AT&T.