Computador é uma máquina que a gente liga na tomada...

- Como isso se conecta com executar um programa ou digitar um texto???

- Pense nisso, traga o que vc conseguir e as dúvidas. Semana que vem começamos com uma discussão rápida sobre este assunto.

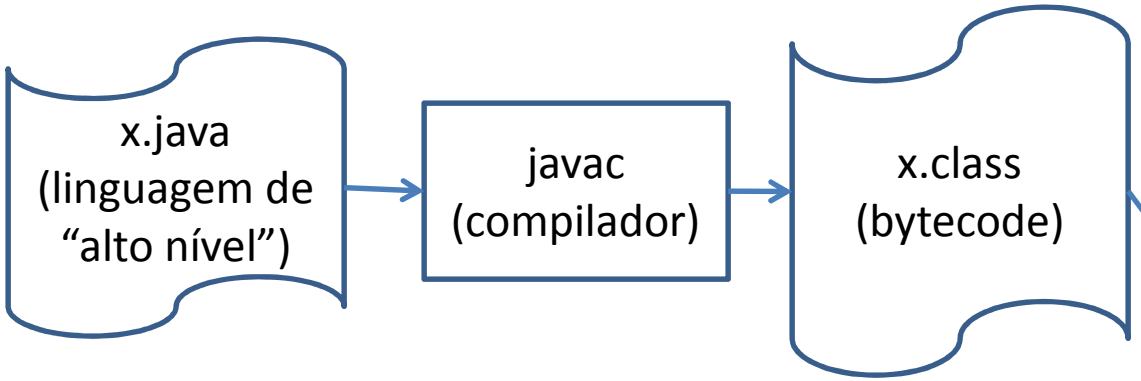
Sumário Aula 3

- Arquitetura de Von Neumann
  - Memória
    - lista endereçada (indexada) que permite acesso “aleatório”.
    - tamanho da palavra
  - Unidade lógica e aritmética
    - Operações aritméticas (+,-,*,/)
    - Operações lógicas (~,&,|)
    - Operações relacionais (>,>=,<,<=,==,!=)
    - Acumulador
  - Unidade de controle
    - Apontador de instruções incrementado sequencialmente a menos de instrução de desvio de fluxo de execução.

Sumário Aula 4

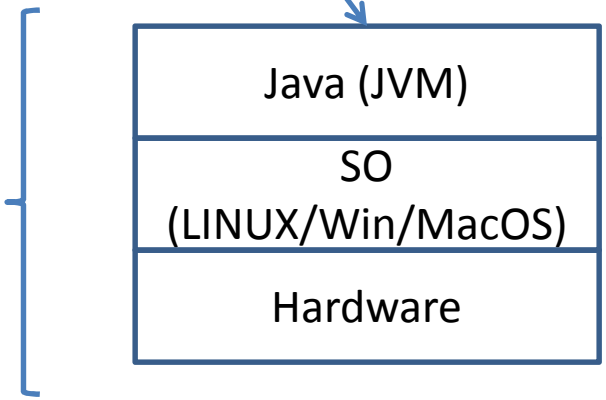
- HIPO
  - Arquitetura de Von Neumann
  - Programa e variáveis armazenadas em memória
  - Exemplo de como instruções são codificadas e executadas em uma máquina.
    - microprocessador/computador
    - máquina virtual
  - Linguagem de máquina (+3120)
  - Linguagem de montagem (assembly) (DNZ 20)

Aula 4 - JAVA

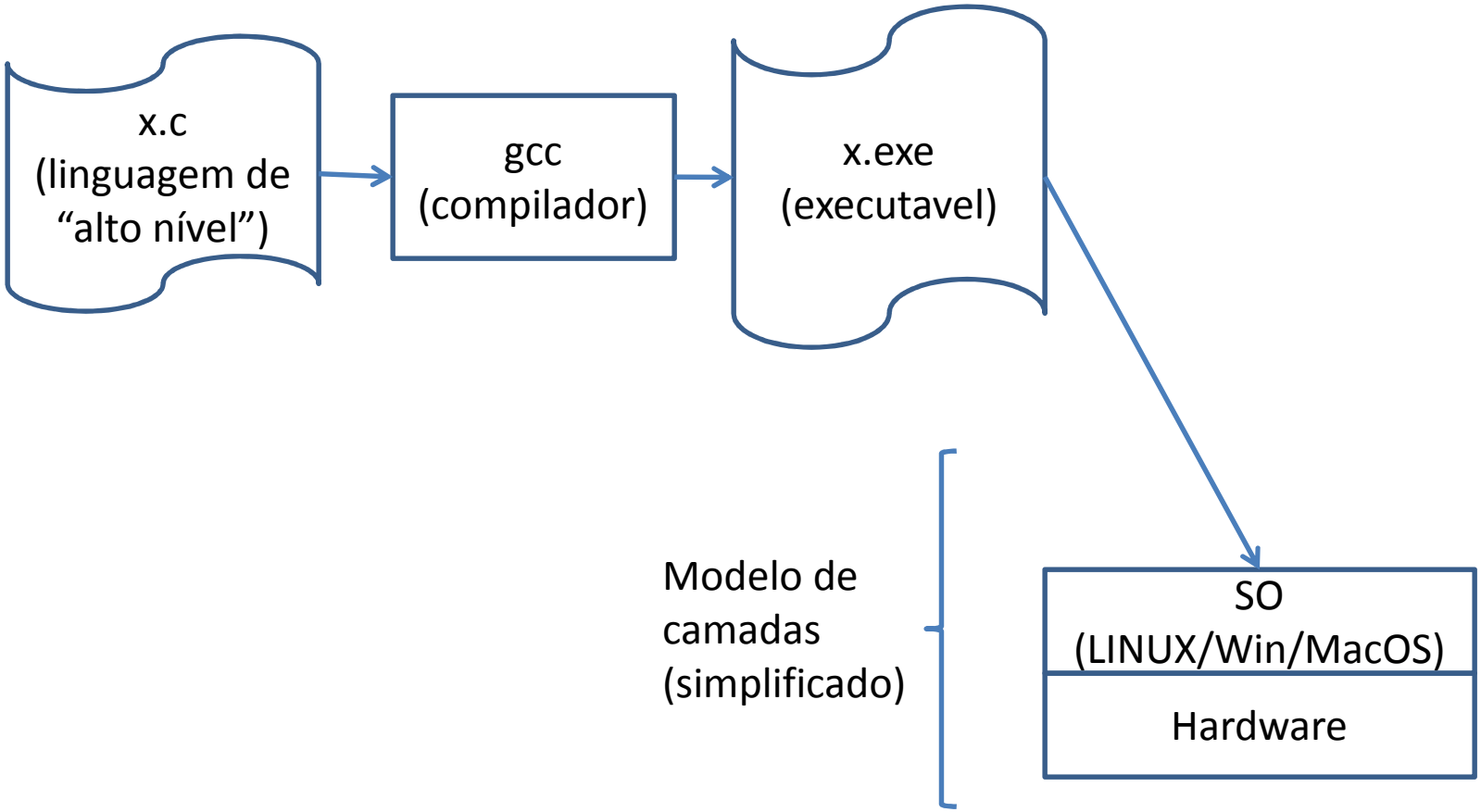


Com essa arquitetura ganhamos comandos que executam ações mais elaboradas e mais “intuitivas” (para seres humanos)

Modelo de camadas (simplificado)



Aula 4 - C


x.c
(linguagem de
“alto nível”)
gcc
(compilador)
x.exe
(executavel)
Modelo de
camadas
(simplificado)
SO
(LINUX/Win/MacOS)
Hardware

Aula 5

Variáveis primitivas

- http://download.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html
- *String não é propriamente primitiva
- **Não há obrigatoriedade de que Boolean seja estritamente representado por 1 bit.

Tipo de Dado	Tamanho (bits)	Representação	Valor default
byte	8	c2	0
short	16	c2	0
int	32	c2	0
long	64	c2	0L
float	32	IEEE754	0.0f
double	32	IEEE754	0.0d
char	16	'\uXXXX'	'\u0000'
String	*	"XXX"	null
boolean	1**		false

Representação para números com sinal

Sinal-magnetude

- Bit de sinal
- Tem que testar o bit de sinal para decidir se soma ou se subtrai.

Complemento de dois(c2)

- Para representar um número **negativo*:** complementar a magnetude em base 2 e somar 1.
- O mesmo hardware pode somar e subtrair os números
- O bit mais significativo é o de sinal.
- Represente zero em complemento de dois

Exercício (não precisa entregar)

Represente em complemento de dois (palavra de 8 bits)

- -30
- 12
- -15
- 57
- -67
- -95
- 128
- -127

Some (palavras de 8 bits em complemento de dois), converta para decimal e confira.

- 00001111 + 10010000
- 00001111+11110001
- 10010101+11111001

- overflow se os dois bits mais significativos do carry forem 10 ou 01.

Böhm e Jacopini 1966

- Teorema da programação estruturada
  - Execução em sequencia
  - Execução condicional
  - Execução até que uma variável seja verdadeira (repetição)
- A linguagem do HIPO satisfaz o teo??
  - Sim, mas é difícil de programar com ela.
- Pode-se criar uma linguagem e traduzí-la para a linguagem do HIPO (ou de qq processador) usando um programa, este se chama compilador.

Controle de fluxo - condicional

```
if (expressão_lógica==true) {
    // bloco então
}
else {
    // bloco senão
}
```

Exemplo

```
double xa= 1.5, ya= 2.5;  // coordenada do ponto inicial (metros)
double xb= 5.7, yb= 6.7;  // coordenada do ponto final (metros)
double comprimentoQ;

comprimentoQ= (xa-xb)*(xa-xb)+(xa-xb)*(ya-yb);
if (comprimentoQ > 49) {
       // bloco então
       System.out.Println ("segmento tem mais de 7m");
}
else {
       // bloco senão
       System.out.Println ("segmento tem no máximo 7m");
}
```

Exercicio em classe

```
if (a==true) {
  // bloco 1
  if (b==true) {
    // bloco 3
    if (c==true) {
      // bloco 5
    }
    else {
      // bloco 6
    }
  }
  else {
    // bloco 4
  }
}
else {
  // bloco 2
}
```

- Apresente, para todas as combinações de valores de variáveis, qual bloco de código será executado.

Controle de fluxo - switch

```
switch (i) {
  case 0: S1= “Tempo seco”;
          break;
  case 1: S1=“Chuva fraca”;
          break;
  case 2: S1=“Chuva moderada”;
          break;
  case 3: c1=“Chuva forte”;
          break;
  default:
          c1=“ERRO!!”
}
```

Controle de fluxo - repetição

```
while (expressão_lógica==true) {
    // bloco a repetir
}
```

Exemplo

```
Boolean Chuva= true;          // Chuva==true significa está chovendo
int t=0;                      // tempo (minutos)
int tParouDeChover=40;        // instante em que para de chover (minutos)
int tfinal=60;                // fim da simulação

while (t <= tfinal) { // o que acontece quando tfinal==0??
      if (t>=tParouDeChover) {
            Chuva=false;
      }
      if (Chuva==false) {
            System.out.Println ("Bicicleta!!!!");
      else {
            System.out.Println ("casa...");
      }
      t++;
}
```

Controle de fluxo - repetição

```
do {
     // bloco a repetir
} while (expressão_lógica==true)
```

Exemplo

```
Boolean Chuva= true;          // Chuva==true significa está chovendo
int t=0;                      // tempo (minutos)
int tParouDeChover=40;        // instante em que para de chover (minutos)
int tfinal=60;                // fim da simulação

do {// o que acontece quando tfinal==0??
       if (t>=tParouDeChover) {
              Chuva=false;
       }
       if (Chuva==false) {
              System.out.Println ("Bicicleta!!!!");
       else {
              System.out.Println ("casa...");
       }
       t++;
} while (t <= tfinal);
```

Controle de fluxo – repetição (contagem)

```
for (int i=0;expressão_lógica==true;i++) {
     //bloco de comandos
}
```

Exemplo

```
Boolean Chuva= true;          // Chuva==true significa está chovendo
int tParouDeChover=40;        // instante em que para de chover (minutos)
int tfinal=60;                // fim da simulação

for (int t=0;t<=tfinal;t++) {  // o que acontece quando tfinal==0??
      if (t>=tParouDeChover) {
            Chuva=false;
      }
      if (Chuva==false) {
            System.out.Println ("Bicicleta!!!!");
      else {
            System.out.Println ("casa...");
      }
      t++;
}
```

Lista 2

- Escreva um programa que toma um inteiro entre 0 e 15 em uma variável e imprima um caracter que representa esse inteiro em hexadecimal.
- Considere que cada inteiro só pode conter um dígito. Use do..while e conte de 00 a 99.
- Faça o mesmo para um número de dois dígitos hexadecimais de 00 a FF.
- Escreva um programa que imprima a representação hexadecimal de um número dado.
- Entrega da tarefa pelo CoL até 25.03.2011.