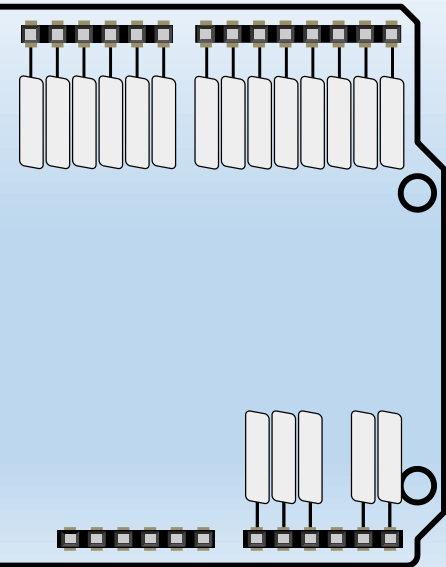


Funções II

Bibliotecas



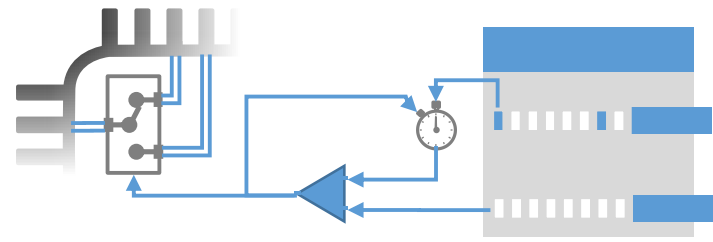
As bibliotecas padrão evitam que os programadores precisem de reinventar a roda.

Bjarne Stroustrup

Passagem de parâmetros

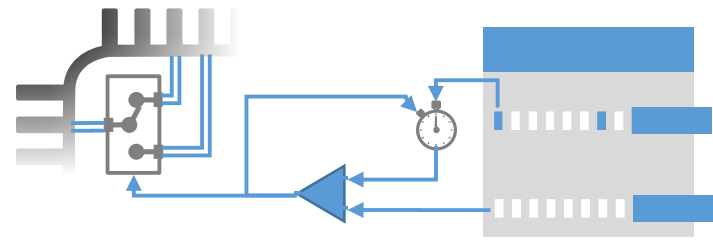
Passagem de parâmetros

- A linguagem C permite dois modos para passar parâmetros entre funções.
 - Por valor.
 - Por referência.
 - Pode não ter parâmetros.



Passagem de parâmetros

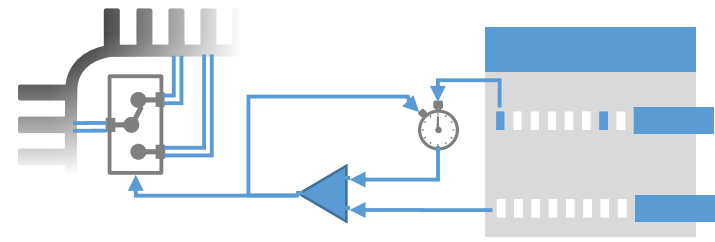
- Passagem por Valor
 - É também conhecida como passagem por cópia.
 - Quando C compila a função e o código que chama a função:
 - A função recebe uma cópia dos valores dos parâmetros.
 - Se o valor de um parâmetro local é mudado, a mudança afeta somente a função e não tem efeito fora dela.



Passagem de parâmetros (valor)

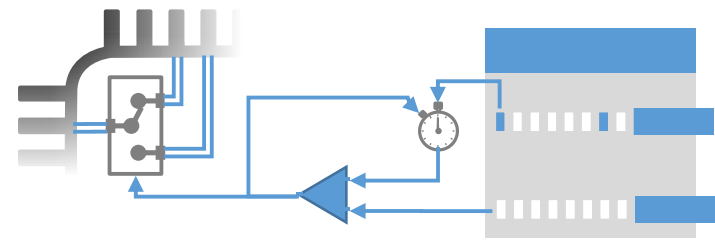
```
void incrementa(int a){  
    a += 1;  
}
```

```
void main(void){  
    int x = 10;  
  
    incrementa(x);  
  
    return 0;  
}
```



Passagem de parâmetros

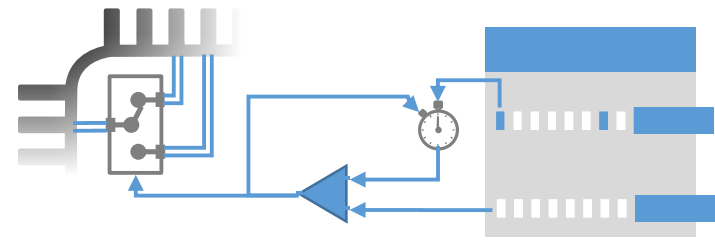
- Passagem por Referência
 - A passagem de parâmetro por referência é usado quando se deseja que uma função modifique o valor do parâmetro passado e devolva o valor modificado para a função de chamada.
 - O compilador passa o endereço de memória do valor do parâmetro para a função.
 - Quando se modifica o valor do parâmetro (a variável local), o valor fica armazenado no mesmo endereço de memória.
 - Ao retornar para a função de chamada, o endereço de memória onde foi armazenado o parâmetro conterá o valor modificado.
 - Para declarar um parâmetro como passagem por referência, o símbolo & deve preceder o nome da variável.



Passagem de parâmetros

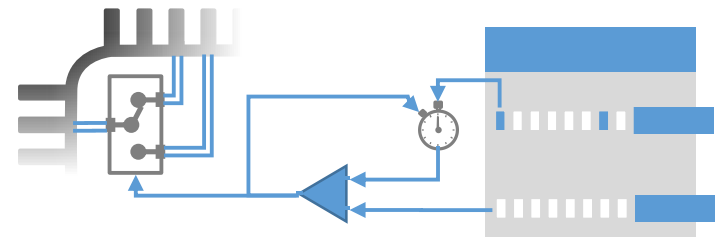
- A passagem de parâmetros por referência permite que uma função “retorne” mais de um parâmetro por vez.
- Isto é feito passando-se uma variável via referência para que a função possa alterar seus valores.

```
//a e b são dados de entrada
//d e r são dados de saída
void div(int a, int b, int* d, int* r){
    (*d) = a / b;
    (*r) = a % b;
    return;
}
void main(void){
    int d, r;
    div(10, 3, &d, &r);
    //d = 3 e r = 1;
}
```



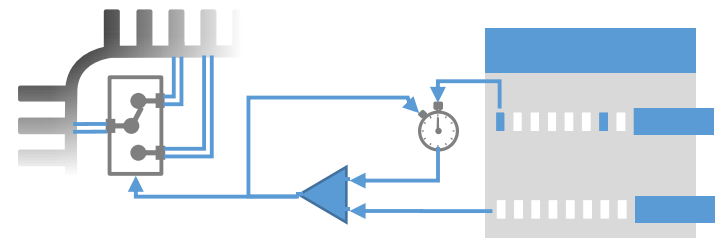
Passagem de parâmetros (referência)

```
void incrementaptr(int* a){  
    (*a) += 1;  
}  
  
int main(int argc, char *argv[]){  
    int x = 10;  
  
    incrementaptr(&x);  
  
    return 0;  
} //end main
```



Diferenças entre valor e referência

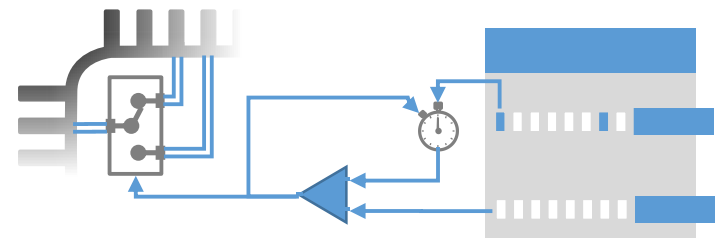
- Parâmetros por valor
 - Os parâmetros por valor são declarados sem & e recebem cópias dos valores dos parâmetros passados.
 - A atribuição de uma função a parâmetros por valor nunca muda o valor do parâmetro original passado.
- Parâmetros por referência
 - Os parâmetros por referência são declarados com & e recebem o endereço dos parâmetros passados.
- As atribuições feitas a parâmetros por referência, mudam os valores dos parâmetros originais.



Bibliotecas

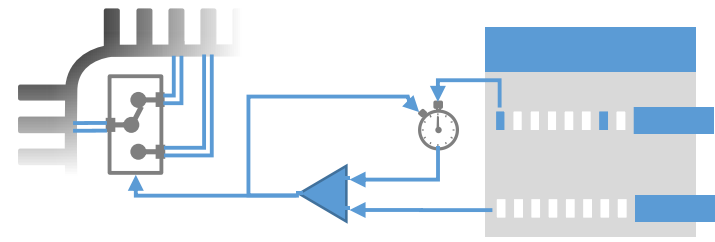
Bibliotecas

- Uma biblioteca é um conjunto de funcionalidades reunidas visando facilitar seu uso.
- Em geral são separadas por tipo de funcionalidade.
- Na primeira versão da linguagem C haviam 15 bibliotecas padronizadas. Em 1995 três novas bibliotecas foram adicionadas. Em 1999 foram seis e em 2011 cinco, totalizando 29 atualmente.
- A disponibilidade destas bibliotecas é dependente da versão do compilador que está em uso.



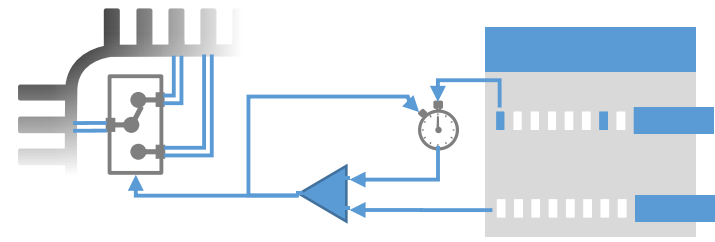
Bibliotecas padrão

- `assert.h`: define a macro `assert()` que implementa uma asserção, utilizada para verificar suposições feitas pelo programa
- `ctype.h`: rotinas para verificação de tipos de dados
- `errno.h`: fornece macros para identificar e relatar erros de execução através de códigos de erro. A macro `errno` fornece um número inteiro positivo contendo o último código de erro fornecido por alguma função ou biblioteca que faz uso do `errno`
- `float.h`: define macros que especificam características do ponto flutuante
- `limits.h`: define macros que definem os limites dos tipos de dados



Bibliotecas padrão

- `math.h`: funções matemáticas
- `stdio.h`: rotinas de entrada e saída definidas pelos criadores da linguagem C
- `stddef.h`: vários tipos de dados e macro substituições
- `stdlib.h`: possui funções envolvendo alocação de memória, controle de processos, conversões e outras.
- `string.h`: fornece funções, macros e definições da biblioteca padrão da linguagem de programação C para manipulação de cadeias de caracteres e regiões de memória
- `time.h`: fornece protótipos para funções, macros e definição de tipos da biblioteca padrão da linguagem de programação C para manipulação de datas e horários de modo padrão.



Biblioteca

- Em geral uma biblioteca é composta de dois arquivos:
 - um arquivo de cabeçalho, ou header, com extensão .H,
 - um arquivo de código, com extensão .C.
- O arquivo .H funciona como um índice, apresentando as funções e definições disponíveis
- A implementação das funções é realizada no arquivo .C, bem como o armazenamento de informações.

