

Programação Orientada a Pilhas

Fundamentos

Prof. Edson Alves

Campus UnB Gama

Sumário

1. Forth
2. Conceitos elementares de Forth

Forth

- ▶ Forth é uma linguagem baseada em pilhas, desenvolvida por Charles H. Moore nos anos 70
- ▶ Segundo o próprio autor, ela foi desenvolvida enquanto ele trabalhava em um IBM 1130, um computador de terceira geração
- ▶ Ele considerou que os resultados que tinha até momento eram tão bons que ele considerou que estava desenvolvendo uma “*fourth-generation computer language*”
- ▶ Daí surgiu a ideia de nomeá-la FOURTH
- ▶ Porém o IBM 1130 permitia identificadores com, no máximo, cinco caracteres
- ▶ Por este motivo ele deu à linguagem o nome FORTH
- ▶ Forth foi utilizada pela NASA no desenvolvimento de aplicações para missões espaciais

GForth

- ▶ GForth é uma implementação *open source* da linguagem
- ▶ Em Linux, ele pode ser instalado com o comando

```
$ sudo apt install gforth
```

- ▶ Para checar se a instalação foi bem sucedida, rode o comando

```
$ gforth --version
```

- ▶ O GForth tem dois modos de operação: o interativo (REPL) e o modo interpretador, onde ele lê e executa as instruções contidas em um *script* Forth
- ▶ Para entrar no modo REPL, basta invocar o interpretador com o comando abaixo

```
$ gforth
```

- ▶ Para encerrar a sessão, utilize a palavra **bye**

Interpretador GForth

- ▶ Para utilizar o GForth no modo interpretador, deve se escrever os comandos Forth em um arquivo de texto (*script*)
- ▶ A extensão adotada para *scripts* Forth é *.fs*
- ▶ Para interpretar um *script* Forth basta invocar o GForth, passando como argumento o nome do *script*

```
$ gforth script.fs
```

- ▶ O *script* abaixo implementa o tradicional “*Hello World!*” em Forth:

```
1 \ A palavra CR imprime uma quebra de linha na saída
2 ." Hello, Forth!" CR
3 bye
```

Palavras

- ▶ Forth organiza seus códigos por meio de comandos nomeados, que abstram tarefas ou ações correlacionadas por meio de um nome comum
- ▶ Estes comandos nomeados seriam os equivalentes a funções em outras linguagem
- ▶ A sintaxe para a criação de um novo comando é

`: nome implementação ;`

ou seja, inicia com dois-pontos, segue com o nome e continua com a definição, que termina com ponto-e-vírgula

- ▶ Comandos definidos dessa forma são denominados **palavras** (*words*)
- ▶ O padrão ANS da linguagem disponibiliza um conjunto grande (mais de 300) palavras pré-definidas, que podem ser usadas para definir novas palavras
- ▶ A habilidade de definir palavras por meio de palavras já definidas é denominada **extensibilidade**

Modo interativo

- ▶ No modo interativo do Forth (REPL), o interpretador responde aos comandos executados de forma bem sucedida com a palavra “ok”
- ▶ Por exemplo, entre no modo interativo e digite a tecla ENTER: Forth responderá “ok”, movendo o cursor para a próxima linha
- ▶ Para inserir um ou mais números na pilha, basta inseri-los, na ordem desejada e separados por um espaço em branco, e digitar ENTER
- ▶ Para visualizar o estado atual da pilha, use a palavra `.s`, a qual não tem parâmetros

```
1 2 3 5 7 11
2 .s CR      \ <5> 2 3 5 7 11, <n> indica o tamanho da pilha
3 bye
```

Palavras para a manipulação do terminal

- ▶ Quando uma palavra recebe um ou mais argumentos, eles são extraídos da pilha, do último para o primeiro
- ▶ Por exemplo, a palavra **spaces** recebe um argumento n e imprime n espaços no terminal

```
1 5 spaces      \ Insere 5 na pilha e, em seguida, imprime 5 espaços no terminal
2 bye
```

- ▶ A palavra **emit** recebe um inteiro n imprime no terminal o caractere cujo código ASCII é n :

```
1 42 emit CR    \ Imprime um asterisco no terminal
2 bye
```


Palavras para a manipulação do terminal

- ▶ O código abaixo define uma nova palavra, chamada **STAR**, que imprime um asterisco no terminal:

```
1 : STAR    42 emit ; \ A convenção é separar o nome da implementação com 3 espaços
2
3 STAR STAR STAR CR   \ Imprime uma nova linha composta de 3 asteriscos
4 bye
```

- ▶ Também é possível definir uma nova palavra, chamada **STARTS**, que recebe um argumento n e imprime n asteriscos consecutivos:

```
1 : STAR    42 emit ;
2 : STARTS  0 DO STAR LOOP ; \ As palavras DO e LOOP serão explicadas adiante
3
4 5 STARTS CR               \ Imprime uma nova linha com 5 asteriscos
5 bye
```

O dicionário

- ▶ Todas as palavras definidas em Forth, seja em biblioteca padrão, seja pelo usuário, são armazenadas no “dicionário”
- ▶ Quando uma nova palavra é definida, Forth compila a palavra e a insere em seu dicionário
- ▶ Por exemplo, a linha abaixo é uma definição alternativa para a palavra **STAR**:
(**[char]** traduz o caractere para seu código ASCII):

```
: STAR    [char] * emit ;
```

- ▶ Quando um comando é inserido no terminal, será ativada a palavra **interpret**, que fará a leitura da entrada em busca de uma string (sequência de caracteres separada por espaços em branco)

O dicionário

- ▶ Se a palavra encontrada consta no dicionário, será ativada a palavra **execute**, que executa a definição da palavra e finaliza com a mensagem “ok”
- ▶ Se a palavra não consta no dicionário, então Forth tentará interpretar a string como um número: caso ele tenha sucesso na conversão, o número lido será inserido na pilha
- ▶ Se a palavra lida não é um número, Forth sinaliza um erro, indicando que a palavra não foi definida
- ▶ Os nomes das novas palavras devem ser compostos por, no máximo, 31 caracteres imprimíveis

Referências

1. **BRODIE**, Leo. [Starting FORTH](#), Online Edition, acesso em 10/10/2025.
2. **HORSE**, M. D. [Learn X in Y minutes - Where X is Forth](#), acesso em 10/10/2025.
3. **MORGAN**, Nick. [Easy Forth](#), acesso em 10/10/2025.
4. **NASA**. [Forth in Space Applications](#), acesso em 10/10/2025.