LINGUAGENS ESOTÉRICAS

BRAINFUCK

LINGUAGENS ESOTÉRICAS

- Linguagens esotéricas são linguagens de programação com uma sintaxe e/ou semântica totalmente malucas
- A comunidade de “linguagens esotéricas” é pequena, os artigos são muito poucos. É um conhecimento extremamente nichado.

HISTÓRIA DAS LINGUAGENS ESOTÉRICAS

- Em 1972, ano quando nasceu a linguagem C, Donald R. Woods e James M. Lyon, ambos de Princeton, criaram uma linguagem chamada “INTERCAL”.
- Na página que Eric Raymond criou alguns anos depois para a linguagem (http://www.catb.org/~esr/intercal/) diz: “Cuidado! Se você não é um hacker hard-core, é melhor ir embora daqui AGORA. Nada além de coisas técnicas malucas e doidas e obsessões sugadoras de cérebro te aguardam além deste ponto! Você foi avisado!”

HISTÓRIA DAS LINGUAGENS ESOTÉRICAS

- A descrição da linguagem por Eric Raymond, e que mostra o espírito da linguagem esotérica, é esta:

Abandona toda a sanidade, quem entra aqui

Então você acha que já viu tudo, né? Ok. Você programou em C, hackeou com LISP. Fortran e BASIC não te amedrontam. Você escreveu modos do Emacs por diversão. Você come Assemblers no café da manhã. Você é fluente em meia dúzia de linguagens que ninguém além de "ubergeeks" ouviram falar. Você ama TECO. Possivelmente até sabe (tremendo) COBOL. Talvez você esteja pronto para o desafio final: Intercal

HISTÓRIA DAS LINGUAGENS ESOTÉRICAS

- Intercal foi criada com um objetivo: ser diferente de absolutamente todas as linguagens de programação da época!
- Essa é conhecida como a primeira linguagem verdadeiramente esotérica
- Intercal mostrou que linguagens com sintaxe e semântica malucas valem a pena se tornar objeto de estudo
- As linguagens esotéricas podem ser criadas com os mais diversos propósitos

PROPÓSITOS DAS LINGUAGENS ESOTÉRICAS

- Minimalismo – Reduzir ao mínimo necessário de instruções possíveis. Ex.: Brainfuck, OISC, Lazy K
- Novos Conceitos – Explorar novos conceitos de design em programação. Ex.: Befunge, Thue e Unlambda
- Esquisitisse – Ser estranha e difícil de programar. Ex.: INTERCAL e Malbolge (quase impossível usar)
- Temática – Baseada em um tema. Ex.: Shakespeare e RockStar

PROPÓSITOS DAS LINGUAGENS ESOTÉRICAS

- Brevidade – Desenvolvidas para terem o menor código possível que implemente um algoritmo específico (“Code Golf”, brincar com isso é “Golf Scripting”). Ex.: CJam, Pyth, GolfScript
- Piada – Criadas como piada. Provavelmente nunca serão usadas. l33t, Emo, Ook!
- Obfuscação – Intencionalmente desenvolvidas para serem muito difíceis de ler. Ex.: Unreadable, Whitespace

ELAS PODEM SER ÚTEIS NO HACKING?

- Linguagens Esotéricas podem burlar filtros e, assim, explorar vulnerabilidades que não seriam possíveis normalmente (ex.: JSFuck).
- Linguagens Esotéricas podem ser usadas para esconder mensagens ou códigos inteiros sem serem detectados (ex.: Whitespace)
- Linguagens Esotéricas podem levar a um entendimento maior sobre como determinada tecnologia funciona (BeFunge, BrainFuck)
- E muito mais

BRAINFUCK

- Uma das linguagens esotéricas mais famosas que existe!
- Criada em 1993
- Turing-completa
- Inspirou várias outras linguagens esotéricas!
- Linguagem Imperativa, como C
- Também conhecida como “brainf***”, “brainf*ck”, “brainfsck”, “b****fuck”, “brainf**k”, “branflakes”, “brainoof”, “brainfrick”, etc

BRAINFUCK

- A linguagem opera em um array de células de memória
- Todas as células eram definidas como zero inicialmente
- A primeira implementação tinha 30.000 “células de memória”. Porém a linguagem em si não tem especificação de memória
- Existe um ponteiro que aponta para a primeira célular de memória inicialmente

BRAINFUCK - COMANDOS

- A linguagem tem apenas 8 comandos!
- > - Move o ponteiro para a direita
- < - Move o ponteiro para a esquerda
- + - Incrementa a célula de memória onde está o ponteiro
- - - Decrementa a célula de memória onde está o ponteiro
- . – Envia o valor da célula para o output
- , - Recebe um caracter e salva na célula de memória onde está o ponteiro
- [- Entra em loop com “]” se a célula atual for diferente de zero
-] – Quando chega neste ponto, volta para “[“ se a célula atual não for zero

BRAINFUCK - PROGRAMANDO

- Portanto os 8 símbolos que funcionam com a linguagem é: ><+-.,[]
- Tudo que for diferente dos 8 símbolos, é ignorado
- Para treinar, use https://fatiherikli.github.io/brainfuck-visualizer
- Um código simples de “Hello World” sem nenhuma otimização:
- ++
  ++++++++++++++++++++++++.++++++++++++++++++++++++++++++.
  +++++++..+++.>++++++++++++++++++++++++++++++++.>+++++++
  ++
  ++++++++++++++++++++++++++++.<<.+++.------.--------.>+.

BRAINFUCK - PROGRAMANDO

- Pedindo para o usuário digitar dois valores e diminuir eles (na forma x-y. Portanto considerando que o primeiro será maior que o segundo):
- ,>,[-<-
  >]<++.

BRAINFUCK - PROGRAMANDO

- Mostrando a Tabela ASCII:
- .+[.+]

BRAINFUCK – DESAFIO 1

- Criar um código em Brainfuck que mostre seu nome completo na tela

BRAINFUCK – DESAFIO 2

- Criar um código em Brainfuck que multiplique dois números de 1 dígito apenas que devem ser informados pelo usuário

BIBLIOGRAFIA

- Não tem! =D


OBRIGADO!