

Linguagens de Programação

Coleta Automática de Lixo

Carlos Bazilio

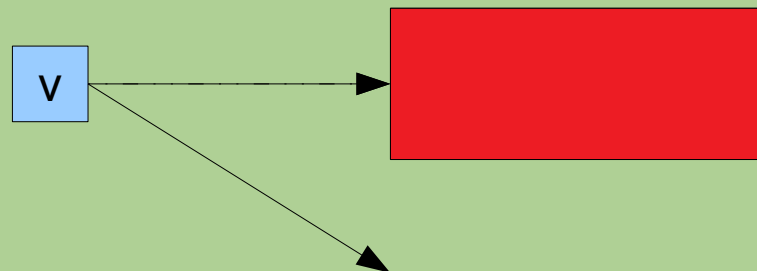
carlosbazilio@id.uff.br

<http://www.ic.uff.br/~bazilio/cursos/lp>

Coleta Automática de Lixo

- Mecanismo pelo qual áreas de memória não mais utilizadas no heap são devolvidas de forma transparente (sem o auxílio do programador) para o S.O.

```
lixo.c x
1 main() {
2     int *v;
3     v = (int *) malloc (sizeof(int));
4     v = NULL;
5 }
```



Coleta Automática de Lixo

John McCarthy (computer scientist)

From Wikipedia, the free encyclopedia

John McCarthy (September 4, 1927 – October 24, 2011) was an American [computer scientist](#) and [cognitive scientist](#). McCarthy was one of the founders of the discipline of [artificial intelligence](#).^[1] He coined the term "[artificial intelligence](#)" (AI), developed the [Lisp](#) programming language family, significantly influenced the design of the [ALGOL](#) programming language, popularized [timesharing](#), and was very influential in the early development of AI.

McCarthy spent most his career at [Stanford University](#).^[2] He received many accolades and honors, such as the 1971 [Turing Award](#) for his

John McCarthy



John McCarthy at a conference in 2006

Born	September 4, 1927 Boston, Massachusetts, U.S.
Died	October 24, 2011 (aged 84) Stanford, California, U.S.

Coleta Automática de Lixo

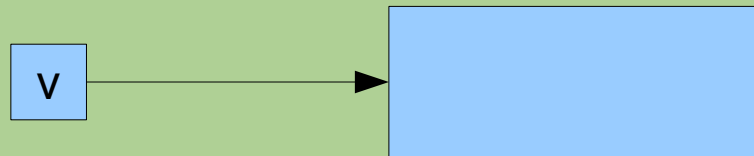
- 2 algoritmos representam boa parte das implementações de coletores de lixos:
 - Contagem de referências e *Mark-Sweep* (Marca e Limpa)

Contagem de Referências

- O algoritmo de contagem de referências é uma técnica básica para implementação da coleta de lixo
- Neste, um contador de referências é mantido para cada área de memória alocada no heap
- A cada alteração do número de referências a uma dada área, o contador é atualizado
- Coletores com estas características costumam ser classificados como gulosos

Contagem de Referências

```
lixo.c x
1 main() {
2     int *v;
3     v = (int *) malloc (sizeof(int));
4     v = NULL;
5 }
```



Mark-and-Sweep

- Este algoritmo pode ser dividido em 3 partes:
 - O coletor marca todas as células como lixo
 - Em seguida, todo ponteiro para o heap é percorrido e as células para os quais estes apontam são marcados como áreas úteis
 - Finalmente, todas as áreas ainda marcadas como lixo são devolvidas para a lista de áreas livres
- Este processo só se inicia após o esgotamento do heap
- Algoritmos deste tipo são chamados de *lazy*