Create account Log in

Article Talk Read Edit More Search

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Polski
Edit links

# Cheney's algorithm

From Wikipedia, the free encyclopedia

> This article includes a list of references, but **its sources remain unclear** because it has **insufficient inline citations**. Please help to improve this article by introducing more precise citations. *(April 2014)*

**Cheney's algorithm**, first described in a 1970 ACM paper by C.J. Cheney, is a method of garbage collection in computer software systems. In this scheme, the heap is divided into two equal halves, only one of which is in use at any one time. Garbage collection is performed by copying live objects from one semispace (the from-space) to the other (the to-space), which then becomes the new heap. The entire old heap is then discarded in one piece.

Cheney's algorithm reclaims items as follows:

- **Object references on the stack.** Object references on the stack are checked. One of the two following actions is taken for each object reference that points to an object in from-space:
  - If the object has not yet been moved to the to-space, this is done by creating an identical copy in the to-space, and then replacing the from-space version with a forwarding pointer to the to-space copy. Then update the object reference to refer to the new version in to-space.
  - If the object has already been moved to the to-space, simply update the reference from the forwarding pointer in from-space.
- **Objects in the to-space.** The garbage collector examines all object references *in the objects that have been migrated to the to-space*, and performs one of the above two actions on the referenced objects.

Once all to-space references have been examined and updated, garbage collection is complete.

The algorithm needs no stack and only two pointers outside of the from-space and to-space: a pointer to the beginning of free space in the to-space, and a pointer to the next word in to-space that needs to be examined. For this reason, it's sometimes called a "two-finger" collector --- it only needs "two fingers" pointing into the to-space to keep track of its state. The data between the two fingers represents work remaining for it to do.

The forwarding pointer (sometimes called a "broken heart") is used only during the garbage collection process: When a reference to an object already in to-space (thus having a forwarding pointer in from-space) is found, the reference can be updated quickly simply by updating its pointer to match the forwarding pointer.

Because the strategy is to exhaust all live references, and then all references in referenced objects, this is known as a *breadth-first* list copying garbage collection scheme.

## Semispace [edit]

Cheney based his work on the *semispace* garbage collector, which was published a year earlier by R.R. Fenichel and J.C. Yochelson.

## Equivalence to tri-color abstraction [edit]

Cheney's algorithm is an example of a tri-color marking garbage collector. The first member of the gray set is the stack itself. Objects referenced on the stack are copied into the to-space, which contains members of the black and gray sets.

The algorithm moves any white objects (equivalent to objects in the from-space without forwarding pointers) to the gray set by copying them to the to-space. Objects that are between the scanning pointer and the free-space pointer on the to-space area are members of the gray set still to be scanned. Objects below the scanning pointer belong to the black set. Objects are moved to the black set by simply moving the scanning pointer over them.

When the scanning pointer reaches the free-space pointer, the gray set is empty, and the algorithm ends.

## References [edit]

- Cheney, C.J. (November 1970). "A Nonrecursive List Compacting Algorithm". *Communications of the ACM*

**13** (11): 677–678. doi:10.1145/362790.362798.

- Fenichel, R.R.; Yochelson, Jerome C. (1969). "A LISP garbage-collector for virtual-memory computer systems". *Communications of the ACM* **12** (11): 611–612. doi:10.1145/363269.363280.
- Tutorial at the University of Maryland, College Park

Categories: Automatic memory management