 Doug Lea's Malloc

\subsubsection{Introducción}

Dlmalloc lleva en uso desde 1987 y es la versión estándar de malloc() y free() en muchos ambientes, por lo tanto su confiabilidad es muy alta.

La “Memory allocator library ” de Doug Lea es una librería estándar que empezó a escribirse en 1987, normalmente se la conoce como dlmalloc.

Dlmalloc es una librería abierta cuyo código puede ser obtenido en <ftp://g.oswego.edu/pub/misc/malloc.c>.

Debido a esto el uso de esta librería es de uso extendido y general. Incluso sirve como la versión nativa default de la implementación de malloc y free en algunas versiones de Linux. Esta librería normalmente hace overwrite al malloc nativo en varios paquetes de software comunes y además se usa en varios ambientes de PC así como en sistemas embebidos.

\subsubsection{Metas}

Estas son las metas propuestas en el desarrollo de la dlmalloc.

Maximizar Compatibilidad, en particular obedecer las convenciones ANSI/POXI

Maximizar Portabilidad, es decir, depender lo menos posible de características dependientes del Sistema como system calls, ademas de conformarse a todos los constraints conocidos del Sistema.

Minimizar el Espacio. Esto se refiere a que el allocator no debe desperdiciar memoria. Debe obtener la mínima cantidad de memoria que sea posible y mantener la misma de tal manera que se minimiza la fragmentación de la misma.

Minimizar el Tiempo. Quiere decir que malloc(), free() y realloc() deben ser lo mas rapidas posibles en cuanto a su tiempo promedio de ejecución.

Maximizar Afanabilidad (Tunability). Las características y comportamientos opcionales de dlmalloc deben poder ser controlados por el usuario, ya sea estática o dinámicamente.

Maximizar Localidad. Se ubican pedazos de memoria que se utilizan típicamente juntos en cercanía unos de otros. Esto ayuda a minimizar la cantidad de miss de páginas y de cache durante la ejecución.

Detección de Errores. Aunque no es posible que esta librería de uso general sirva también como una librería de detección de errores de memoria como *Purify*, si debe proveer algún medio para detector corrupción de memoria debido a sobre-escritura, múltiples frees, etc..

Minimizar Anomalías. Si el allocator es configurado usando los valores default debería funcionar bien en una amplia gama de aplicaciones tales como aplicaciones GUI, compiladores, interpretadores, IDEs , programas con intensive networking, paquetes gráficos, web browsers , etc …

\subsubsection{Algoritmos}

La Dlmalloc usa dos algoritmos como base para su funcionamiento. Boundary y Binning, los cuales han permanecido aproximadamente iguales desde sus primeras versiones.

Boundary Tags(Tags de frontera)

Chunks of memory carry around with them size information fields both before and after the chunk. This allows for two important capabilities:

* Two bordering unused chunks can be coalesced into one larger chunk. This minimizes the number of unusable small chunks.
* All chunks can be traversed starting from any known chunk in either a forward or backward direction.

Los pedazos de memoria llevan consigo el size tanto al principio como al final del mismo, lo que permite la implementación de dos capabilities muy importantes.

* Dos pedazos de memoria contiguos y sin utilizar pueden juntarse en un pedazo más grande, esto minimiza la fragmentación y el número de pedazos pequeños inutilizables.
* Se puede viajar a través del pedazo de memoria en cualquier dirección.

\includegraphics[height=14cm]{imagenes/boundary.png}

Binning

Los pedazos de memoria disponibles se agrupan en contenedores, los mismos agrupados por tamaño. La búsqueda de pedazos disponibles se realiza buscando el pedazo más pequeño que mejor le quede al requerimiento del proceso. (smallest-first, *best-fit* order).

\includegraphics[height=14cm]{imagenes/binning.png}

En las versiones más recientes los pedazos se ordenan dentro de los contenedores, se ordenan por tamaño y por edad.

Entonces en general los pedazos libres se juntan con sus vecinos libres y se meten en un contenedor que los ordena y los dispone según se requiera.

Thus, the general categorization of this algorithm is *best-first with coalescing*: Freed chunks are coalesced with neighboring ones, and held in bins that are searched in size order.

En sistemas operativos de 32 bits el pedazo de memoria más pequeño que se puede ofrecer a la memoria virtual es de 16 bytes, en los de 64 bits el más pequeño es de 24 bytes.

La implementación uso de este algoritmo es bastante rápida. Aunque lo mostrado arriba son las características básicas, muchas otras pequeñas y grandes adiciones al dlmalloc se han hecho a través de los años para mejorar aún más la implementación.

CONFIABILIDAD

DLmalloc es un memory allocator muy confiable. Se empezó a usar en 1987 y se ha ido mejorando a través de los años. Es el estándar en muchos ambientes

DEBUGGING

El debugging no es recomendable a no ser que el programador lo requiera en alguna aplicación específica.