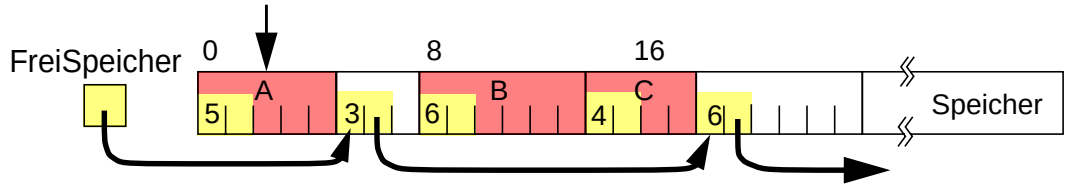
Aufgabe 3:

06.05.2004

halde (12 Punkte)

In dieser Aufgabe soll eine einfache Freispeicherverwaltung implementiert werden, welche die Funktionen **malloc(3C)**, **calloc(3C)**, **realloc(3C)** und **free(3C)** aus der Standard-C-Bibliothek ersetzt. Die Freispeicherverwaltung soll den freien Speicher in einer einfach-verketteten Liste verwalten. Am Anfang eines freien Speicherbereiches steht jeweils eine Verwaltungsstruktur mit der Größe des Speicherbereiches und einem Zeiger auf den nächsten freien Bereich. Ein globaler Zeiger zeigt auf den Anfang der Liste.

Im Verzeichnis */proj/i4sos/pub/aufgabe3/* befinden sich die Dateien *halde.c, halde.h, wsort.o* und *bug.c* sowie ein passendes *Makefile*. Kopieren Sie sich die Dateien in Ihr Arbeitsverzeichnis und implementieren Sie die fehlenden Funktionen und Definitionen in *halde.c*.



a) Speicher initialisieren, belegen und freigeben (Version 1)

Implementieren Sie die Funktionen **malloc** und **free**. Beim ersten Aufruf von **malloc** ist vom Betriebsystem ein Speicherblock von 1 Megabyte anzufordern (**sbrk(2)**). Der komplette Block wird in die Freispeicherliste eingefügt. (Ein Nachfordern von mehr Speicher vom BS ist in dieser Aufgabe **nicht** vorgesehen.) Die Funktion **malloc** sucht den ersten freien Speicherbereich in der Freispeicherliste welcher groß genug ist, um den geforderten Speicherbedarf plus Verwaltungsstruktur zu erfüllen. Ist der Speicherbereich mind. 10 Byte größer als benötigt so ist der nicht benötigte Speicher wieder in die Freispeicherliste einzuhängen. Dem belegten Speicherbereich ist eine Verwaltungsstruktur vorangestellt. Diese enthält die tatsächliche Größe des belegten Speicherbereichs und eine Kennung mit dem Wert *0x00beef00*. Der zurückgelieferte Zeiger muss hinter die Verwaltungsstruktur zeigen, wie in der Abbildung für den Bereich A gezeigt. Die Funktion **free** hängt einen freigegebenen Speicher wieder in die Freispeicherliste ein. Vor dem Einhängen ist die Kennung zu überprüfen, besitzt diese nicht mehr den Wert *0x00beef00*, so soll das Programm abgebrochen werden **abort**(3C).

b) Speicher freigeben vergrößern und verkleinern (Version 2)

Nun sollen noch die Funktionen **realloc** und **calloc** implementiert werden (**memcpy(3C)**, **memset(3C)**). Diese Erweiterung soll als eigene Version im *RCS* verwaltet werden. Legen Sie hierfür ein Verzeichnis *RCS* im Verzeichnis *src* an und checken Sie Ihre Quellen aus Teilaufgabe a als Version 1.x ein. Die Funktionen aus Teilaufgabe b sollen als Version 2.x eingefügt werden.

c) Debuggen

Debuggen Sie nun das Programm **bug** und beschreiben Sie die Fehler, die Sie finden, in der Datei *halde.txt*. Benutzen Sie zum Debuggen einen Debugger wie z.B. den gdb. Von besonderem Interesse sind Fehler, die zu einem Fehlverhalten in Ihrer Freispeicherverwaltungen führen.

Hinweis zur Lösung dieser Aufgabe:

- Die Funktionen malloc, free, realloc und calloc müssen das in den Manpages beschriebenen Verhalten aufweisen. Denken Sie auch an das Setzen der **errno** im Fehlerfall!

Abgabe: bis spätestens Mittwoch, 19.05.2004, 15:45 Uhr