Implement System Calls

The system call sbrk( ) is used for changing the size of the data segment of a program. It either increases or decreases the size of the heap space during the runtime of the program. This is a low level memory management function, and it is still used in some place by some people. It is used in this example program to create a simple operating system procedure for allocating and deallocating memory.

Objective: To create a simple C program that manually allocates and deallocates memory using the sbrk() system call.

Code Example:

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
int main() {
void *initial_brk, *new_brk;
int increment = 1024; // Allocate 1 KB
// Get current end of data segment
initial_brk = sbrk(0);
printf("Initial program break: %p\n", initial_brk);
// Increase data segment size
new_brk = sbrk(increment);
if (new_brk == (void *)1) {
perror("sbrk increment failed");
return 1;
}
printf("Program break after increment: %p\n", sbrk(0));
// Use the new memory
char *allocated = (char *)new_brk;
strcpy(allocated, "Hello from heap!");
printf("Stored string: %s\n", allocated);
// Decrease data segment size
if (sbrk(increment) == (void *)1) {
perror("sbrk decrement failed");
return 1;
}
printf("Program break after decrement: %p\n", sbrk(0));
return 0;
}
```