CENG311 - HOMEWORK 2

Deadline: 02.12.2022 23:59

In this homework you are supposed to write MIPS assembly instructions where you can create a simple song list. You should use dynamic arrays and implement given methods properly. The user of the program should be able to:

- add a song to the list by providing its name and duration
- delete a song from the list by providing its name
- list all songs

Your implementation should satisfy these constraints below:

- You should use 12 bytes dynamic memory space as a dynamic array for storing songs. First 4 bytes are for capacity of the dynamic array, second 4 bytes are for size of the dynamic array, last 4 bytes are for address of the elements.
- For each song you should allocate 8 bytes space where first 4 bytes are address of the song name (name itself will be 64 bytes) and last 4 bytes are for duration of the song.
- In the subroutine initDynamicArray, you should create an array with size of 2 and store the address in the dynamic array's address of the elements part.
- When the user chooses to add a new song, you should create that song in the heap by using syscall with the sbrk code 9 in createSong subroutine. You should put this song address into the songs by using the putElement subroutine. In the putElement subroutine, you should increase the size of the elements array and put the newly added element's address into the elements array.
- When the user chooses to delete the song, you should find it by using findSong subroutine where you should take the name of the song from the user and use removeElement to remove its address from the songs. In the findSong subroutine you should call another subroutine called compareString to check the name of the songs.
- If an element is removed from the dynamic array, other elements that follow the deleted element should be shifted to the previous empty space (in a dynamic array with 5 elements, if the 3rd element is removed, 4th and 5th elements should be shifted to the 3rd and 4th positions).
- Whenever the size of the dynamic array reaches capacity, you should increase the capacity to 2 times the old one and copy the elements of the elements array into the new allocated elements array. You should assign the value zero as an address for array elements that haven't pointed to any valid song yet. All these operations should be implemented in the putElement subroutine.
- Whenever the size of the dynamic array drops down to capacity/2 - 1, reduce capacity by factor of 2 and allocate space for that capacity unless the size is 2, copy the values of the elements array to the newly allocated elements array. All these operations should be implemented in the removeElement subroutine.
- While you list the songs you should use the listElements subroutine to list the songs. In the listElement subroutine. In this subroutine you should call another subroutine called printElement. You should use printElement subroutine as a transition where you call printSong subroutine.

Important Notes:

- You should not change the template and only fill the empty subroutines initDynamicArray, putElement, removeElement, listElements, compareString, printElement, createSong, findSong, and printSong.
- initDynamicArray, putElement, removeElement, listElements, and compareString subroutines should be generic (can be used for not only songs but also other types).
- You should not use the variables sReg and songListAddress in your instructions.
- All the variables needed given already. Therefore, you should not add any new variable.
- You should use copmStr variable to get a string from user for delete operation. Do not allocate new space for getting the string. Be careful, you should allocate a new 64 bytes space for the name of a song which will be created.
- S registers are expected to remain unchanged after the subroutines are used.
- There is no free heap option in Spim, so you don't have to free any unused heap memory.

Subroutine usages:

Subroutine Name	Argument 1 ($a0)	Argument 2 ($a1)	Argument 3 ($a2)	Return Value ($v0)
initDynamicArray	-	-	-	Address of the dynamic array
putElement	Address of the dynamic array	Address of the song	-	-
removeElement	Address of the dynamic array	Index of the song (-1 for no element)	-	-
listElements	Address of the dynamic array	-	-	-
compareString	Address of the first string	Address of the second string	Comparison size	1 for found 0 for not found
printElement	Address of the song	-	-	Address of the song
createSong	-	-	-	Address of the song
findSong	Address of the dynamic array			Index of the song (-1 for no element)
printSong	Address of the song	-	-	-

Submission details:

- You are supposed to submit a .asm file named <your_id>_hw2.asm
- No collaboration is allowed.
- Use the given template and fill the necessary fields.
- You are allowed to use additional subroutines for your own usage below the given label "additionalSubroutines" (You don't have to).
- Your assignment will be checked in QtSpim simulator. Thus, you should be certain that your instructions work well in QtSpim.