Assignments

Assignments

Assignment 01 - Refresher on C programming - Returned

Title	Assignment 01 - Refresher on C programming
Student	Bosen Yang
Submitted Date	Sep 12, 2018 11:22 pm
Grade	**96.0 (max 100.0)**
History	Wed Sep 12 11:22:48 EDT 2018 Bosen Yang (by570) submitted

Instructions

Assignment 01 - Refresher on C programming

Objectives

1. Get a hang of the programming and compilation tools that will be used throughout the semester.
2. Acquire good habits for assignment submission.

Foreword

This first assignment was deliberately designed to be very easy to solve. If you finish before the end of the session, all the better. If you have trouble answering all the questions in the allotted time, then it is imperative that you brush up your C development skills.

Submission Format

When submitting your work, please stick to the conventions described below.
You will prepare a working directory that will contain the following:
- `bin` directory: empty directory for all the executable files created by editing the links between your object files
- `include` directory: directory containing all of your header files (* .h headers)
- `lib`: directory containing all of your pre-compiled libraries (* .a)
- `obj` directory: empty directory for all the object files (* .o) created by compiling your source files
- `src`: directory containing all of your source code files (*.c)

- `Makefile`: file containing all of your compiler directives
- `README` file: plain text file that reports on your work submission

Of all the elements in the working directory, the `README` file is the most important. It shall contain:
- Your first name, your last name, and your NYU ID
- The detailed list of files you submitted in the directories `include`, `lib`, and `src` (for each file, list the questions it provides a solution for)
- The explanation of the compilation rules provided in your `Makefile`
- Comments about the work you submitted, if any (what does not run, what is encoded but produces errors at compile / execution, what feature is incomplete, ...)
- Textual answers to some questions in the set (eg. Q7 in this set)

Before submitting your working directory, please archive it in TAR format and compress it in GZIP format. The resulting file will be renamed like so:
`[lastname-firstname-nyuid].osc.[#assignment].tgz` (*eg. `marin.olivier.ogm2.osc.01.tgz`*)
This is the file you attach to your submission ***before*** the deadline.

Question Set

Question 1

Extract the file attached to this assignment, and compile/run the program by calling the make command in the terminal.

Question 2

Edit the Makefile to archive the array implementation of the stack as a library.
The goal is to archive the stack_array.o object file in a library libstack.a
The compilation of the final executable must now be carried out without using the object file stack_array.o

Question 3

We want to change the size of the stack at compile time, via the definition of a constant STACK_SIZE contained in the code.
Use gcc -D directive to change this value from the Makefile, and then get its display in the main program of stack_test.c

Question 4

Using an array is not satisfactory; we would prefer to use a doubly linked list.
Complete the list_impl.c file code to:
- Extract an element from the head (the associated cell is removed from the list and the memory it occupied is deallocated)
- Extract an element from the tail (the associated cell is removed from the list and the memory it occupied is deallocated)
- Compute the number of items in the list

Question 5

Write a file called stack_list.c that uses the primitives from list.h to build a dynamic stack that implements stack.h
Add a compiler directive in the makefile to build a new library libstack.a from stack_list.c and list_impl.c
Recompile an executable from the stack test program (stack_test.c) and use your new library to verify that it works correctly.

Question 6

Write a file called fifo_list.c that uses the primitives from list.h to build a dynamic queue that implements fifo.h
Add a compiler directive in the makefile to build a new library libfifo.a from fifo_list.c and list_impl.c
Recompile an executable file from the test program (fifo_test.c) and use your new library to verify that it works correctly.

Question 7

Traversing the entire list to determine the number of items is too expensive. What changes should you make, and in which file(s), to determine the size of the list in O(1)?

Additional resources for assignment

- assignment01.start.tgz (10 KB; Jul 24, 2018 3:33 pm)

Submitted Attachments

- yang.bosen.by570.osc.01.tgz (8 KB; Sep 12, 2018 11:22 pm)

Additional instructor's comments about your submission

%%%% Bosen

Q1
Good.

Q2
Good.

Q3
Good.

Q4
Good.

Q5
Good.

Q6
You miss something while compiling fifo_test.

Q7
Good.

Back to list

Timezone: PRC

- Terms of Use
- Send feedback to the NYU Classes Team
- Powered by Sakai
-