**CS2106 Introduction to Operating Systems**

**Lab 1 - Leveling Up on C**

**Answer Book**

Please read the instructions in the main lab sheet before completing this document. Submission deadline is **Sunday 11 February 2024, 1 pm (1300 hrs)**.

|  |  |
| --- | --- |
| **Student 1** | |
| Name: |  |
| Student ID (AxxxxxxY): |  |
| Group (Bxx): |  |
| **Student 2** | |
| Name: |  |
| Student ID (AxxxxxxY): |  |
| Group (Bxx): |  |

Section 1.

**Question 1.1 (1 mark)**

The #include <stdio.h> statement tells the preprocessor to look for the stdio.h file in the standard system directories. These directories are predefined by the compiler and often include locations like /usr/local/include or /usr/include.

**Question 1.2 (1 mark)**

these variables are meant to be local to the file and are not intended to be used externally by other parts of the program

**Question 1.3 (1 mark)**

The compiler encountered calls to functions (enq and deq) for which it couldn't find explicit declarations.

**Question 1.4 (1 mark)**

void enq(double data);

double deq();h

Section 2

**Question 2.1 (1 mark)**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Global / Local** | **Address** |
| p1 | G | 0x55ae8f802018 |
| p2 | G | 0x55ae8f802020 |
| p3 | G | 0x55ae8f802028 |
| p4 | G | 0x55ae8f802030 |
| w | L | 0x55ae8f802038 |
| x | L | 0x7ffc27e3b1ec |
| y | L | 0x7ffc27e3b1e8 |
| z | L | 0x7ffc27e3b1f4 |

**Question 2.2 (1 mark)**

|  |  |
| --- | --- |
| **Variable** | **Location (S, D, T or H)** |
| p1 | D |
| p2 | D |
| p3 | D |
| p4 | D |
| w | D |
| x | S |
| y | S |
| z | S |

How I inferred these answers from Q2.1:

Pointers (p1, p2, p3, p4) are global pointers, which have static storage duration.

Global variables with static storage duration are stored in the data segment.

w is declared as static within the function fun1.

Static variables, especially those defined inside functions, are stored in the data segment.

Local variables (x, y, z inside fun1):

Local variables within a function, like x, y, and z in fun1, are allocated on the stack.

**Question 2.3 (1 mark)**

Being static, w retains its value between calls to fun1 because its memory space in the data segment. They are created when the program starts and are only destroyed when the program terminates.

**Question 2.4 (1 mark)**

Local Variable (Declared as Static):

* Scope is limited to the function.
* Internal linkage, not visible outside the file.
* Retains value between calls, lifetime throughout program.

Global Variable (Declared as Static):

* Scope is limited to the file.
* Internal linkage, not visible outside the file.
* Retains value between calls, lifetime throughout program.

**Question 2.5 (1 mark)**

Declared acc as a static variable inside the accumulate function.

This modification ensures that acc retains its value between function calls without introducing new variables or using global variables.

Section 3

**Question 3.1 (1 mark)**

Local variables (x, y, z) are allocated on the stack, and their addresses are within a specific range of the stack memory.

Memory returned by malloc is allocated on the heap, a separate region of memory with a different address range.

**Question 3.2 (1 mark)**

p->name = (char \*) malloc(strlen(name) + 1);

dynamically allocates memory of the correct size for the name field, allowing it to store strings of varying lengths and preventing buffer overflows.

free(node->name);

Before freeing the entire TPerson structure, ensure that the dynamically allocated memory for the name field is freed.

**Question 3.3 (1 mark)**

gcc -g testlist.c llist.c -o testlist

valgrind ./testlist

Section 4

**Question 4.1 (1 mark)**

Renaming a file involves updating the hash table to reflect the new filename. If I simply perform strcpy(node->filename, new\_filename), I will change the name of the file within the linked list node, but the hash value associated with the file's name won't be updated. This means that the file may no longer be in the correct linked list after renaming.

**Question 4.2 (1 mark)**

Procedure to rename a file:

If the old file is found Then

Make a new file with the new name

Remove the old file

Else

Show an error message

End If

**TOTAL: \_\_\_\_\_\_\_\_\_\_\_ / 14**