hw02-exercises

Author’s name: Cuiting Huang

Author affiliation: Northeastern University

City & country location: Zhengzhou, China

E-mail address: huang.cui@northeastern.edu

Abstract* – *This is an exercises of the Homework - Module2. In this part 3, I will do my written exercises. In this homework, I have built a ring buffer queue and a linked stack. Here are some usage of these.

Index Terms – Queue, Stack, Pointer, Ring buffer.

I. Question One

Circular queues are used quite a bit in operating systems and high performance systems, especially when performance matters. Do a little outside research, and edit this section of the readme answering specifically: *Why is a ring buffer useful and/or when should it be used?*

Answer

First there are several things we must know about the Circular queue. A circular buffer starts out empty and has a set length, and uses FIFO(first in, first out) logic. Also it has a property that when it is full and a subsequent write is performed, then it starts overwriting the oldest data. When there is a lot of process and data need to be run,we can not store all the data. Our computer processes a data and release it, then process the next one. It will cause the memory wasted of the data that has been processed.

Circular buffers can help us solve the problem completely. It will use only one fixed piece of memory. It help us to avoid creating, canceling, and allocating working memory frequently. We could use less memory and do more processes. Static data storage is used more often in the embedded system development than dynamic allocation.

Here is an example I read the materials from the outside sources. We build a system which needs to operate multiple terminals and monitor their health status in real time. We can use the Circular buffers to achieve this. We set up a socket communication between the server and the terminal, so that the server can maintain a long connection with each terminal, which enables us to realize the server can operate the terminal and receive the health status packets of the terminal[1] .

Circular buffers are also a useful construct for situations where data writes and reads occur at different rates: the latest data is always available. If the speed of reading cannot keep up with the speed of writing, the old data will be overwritten by the new data. By using a circular buffer, we can ensure that we are always using the latest data.

[1] https://www.embdded.com/ring-buffer-basics/

II. Question Two

We are going to talk about stacks quite a lot in this course, so it will be important to understand them. Do a little outside research, and edit this section of the readme answering specifically: Why is a stack useful and/or when should it be used?

Answer

A stack has two main principal operations, PUSH and POP, and it uses LIFO(last in first out) logic. If the stack is full and does not contain enough space to accept an entity to be pushed, the stack is then considered to be in an overflow state. The pop operation removes an item from the top of the stack[2]. There are two storage structures of stack, sequence stack and linked stack.

A linked stack is used to represent a stack to facilitate the insertion and deletion of nodes. When multiple stacks are used simultaneously in the program, a linked stack can not only improve efficiency, but also achieve the purpose of sharing storage space.

Here are some usage scenario of the stack. When we try to call of a subroutine, the address of the next instruction will be stored in the stack then jumping to the subroutine. And when the subroutine is completed and the address will be removed to return to the original program. Calling a recursion routine is very similar with the subroutine. Also we can use a stack to make a traversal of binary trees and Compute an expression. The stack is used in the second and third part of the compiler's front end called the semantic analyzer and the syntax analyzer.

[2] https://en.wikipedia.org/wiki/Stack_(abstract_data_type)