

Simple Data Structure

A simple data structure framework
implemented in C

StarNight @ COSCUP 2013

Who am I?

潘建宏 (StarNight)

mail : zack_pan@alumni.ncu.edu.tw

出沒在～

PTT : zack2004

plurk : StarNight

Facebook : Jian-Hong Pan

GitHub : starnight

現在研替剛結束，
暫時繼續在種花店當個打雜園丁～



Outline

1. What is it?
2. Why do this?
3. How to use it?
 - a. Structure Types
 - b. Dedicated Functions
 - c. Public Functions
4. Demo

What is Simple Data Structure?

- You can have it from the GitHub (include Wiki):

<https://github.com/starnight/simple-data-structure>

1.0.0 Release now!!!

- A "simple" data structure framework implemented in C
- Implemented **QUEUE**, **STACK** and **RING** (Circular Buffer)

Why do Simple Data Structure?

1. We have to use the queue, stack ... data structures in embedded system with limited resources.
2. There are native defined templates, classes, interface or functions in C++, .NET, Perl, PHP, Python and Java
3. But, C does not.

沒有喜歡的，就自幹一個！！！！

by ~~—慣C—~~ 大神s

So, code it by myself in C

Structure Types

```
struct _SDS_BUFFER {  
    uint8_t type;        /* The data structure type. */  
    uint8_t len;         /* The buffer array length. */  
    uint8_t inpos;       /* The next input position. */  
    uint8_t outpos;      /* The next output position. */  
    void *elems;         /* The pointer of buffer array. */  
};
```

SDS_QUEUE, ***SDS_STACK*** and ***SDS_RING*** is
_SDS_BUFFER.

SDS_BUFFER

type
len
inpos
outpos
*elems

Buffer Array in Memory

element #0
element #1
element #2
element #3
element #4
.
.
.
element #len-1

**Could be any
data type array**

Dedicated Functions

- **SDSInitQueue, SDSInitStack, SDSInitRing**
- **SDSPushQueue, SDSPushStack, SDSPushRing**
- **SDSPopQueue, SDSPopStack, SDSPopRing**
- **SDSFrontQueue, SDSTopStack, SDSFrontRing**
- **SDSBackQueue, SDSBackStack, SDSBackRing**

Public Functions

- SDSEmpty: Data structure is empty or not.
- SDSSize: Number of elements in the buffer.
- SDSPush: Push an element into the buffer.
- SDSPop: Pop the first ordered element from the buffer.
- SDSFront: Access next element.
- SDSBack: Access last element.

Demo

1. Have a SDS_QUEUE data structure.
2. Define the length of the buffer.
3. Have an integer buffer array.
4. Initial the queue with assigned length and the buffer array.
5. Use it with SDSPush/Pop/Front/Back functions.

Demo codes:

<https://github.com/starnight/simple-data-structure/#quick-start>



End & Thanks ~