4/7/2022

# define

- NN bậc cao
- Compiler process: main.c → main.exe

+ Source: .h /.c header/source

+ Preprocessor: main.c ↓ main.i → + Source: .i
- Copy tất cả hàm, biến thay thư viện gộp 1 file
- Xoá cmt

Compiler ↓

Assembly code. .i → .s $\xrightarrow{\text{Assembler}}$ object file (.o /obj)

Linker ↓

executable .exe, .dll

5/7/2022

Macro

#define ở đâu có thì thay vào

ex: ~~main~~ source.h → source.i

#define MAX 100 → ở đâu có Max thay 100

#define SUM(a,b) a+b

"≠ gcc code"

Đi chung
#ifdef SIZE
→ K° hiểu ⇒ thêm #define SIZE = 100
#endif

#if SIZE >200   nếu size >200
    int arr[300]   tạo mảng 300
# elif SIZE == 20
    int arr[100]
# else
    int arr[200]

* #undef SIZE   xoá định nghĩa để có thể thay thế

    ## nối chuỗi

#include <stdarg.h> → viết hàm nhưng ko biết bao nhiêu parameter truyền vào

7/7/2022

uint16_t.
tên biến
~~6byte~~ bit

ifndefine_ kiểm xem có ai đã define_ chưa?
Extern

- Tìm tất cả các file xem file nào đã khai báo biến đó!

extern int i;

extern void count(); extern hàm từ cái file khác

* Có thể cục bộ or toàn cục
* K° gán gtrị cho extern (extern int i=10 ✗)

Union sd cái có dung lg lớn nhất
                    size

Ứng dụng Union : listdata

11/7/2022

Các biến đặc biệt (continue)

* biến static

Static int i=0; (tồn tại khi ra khỏi hàm)

+ Đc khởi tạo 1 lần

+ Lần 1 gọi hàm có biến thì sẽ khởi tạo, các lần sau sẽ bỏ qua

* inline

13/7/2022

Lệnh goto

!nguy hiểm khi xài
sd để thoát ra cái
vòng lặp phức tạp

gặp lệnh nhảy xuống luôn

```
goto sole;
printf("...");
sole:
  printf("  ");
  return;
```

try catch in c, c++
+ define try catch

```
try {
  int a=5;
  int b=0;
  int c;
  c=a/b
} catch() {
  printf("Mau = 0");
```

```
#include <setjmp.h>
  biến  jmp_buf env;
        kiểu dữ liệu
```

14/7/2022

Uint 32_t , uint 16_t

Bit Mask

typedef enum
{ GPIO_PIN_RSET = 0
=1 ← GPIO_PIN_SET
} GPIO_PIN_state

0 → 1 → 2 → 3 ...

lệnh assert dùng để debugg

15/7/2022

exit(0);

atexit(test); (gặp nhiều)
Khi KT chương trình sẽ chạy hàm (test) cuối cùng ~~q~~ trong dùng chạy ts số cuối cùng để lưu trữ lại giá trị mong muốn lại trong flash

(*) longjmp(env,1)

18/7/2022

Các hàm thg dụng trong th <stdlib>

* hàm atoi → integer   chuyển kdl string → int

atof → float.   g trị để đc chuyển đổi → chuỗi ↑

* hàm ~~strtod~~ strtod (const char* str, char ** endptr)

→ tham chiếu → 1 đối tg đã đc cấp phát của biến char*, có giá trị đc thiết lập bởi hàm tới ký tự kế tiếp trong str sau g trị số

* strcpy(copy chuỗi đc chỉ đến bg src to dest

char * strcpy (char ~~d~~dest, const char *src)

19/7/2022

bt viết * strtok gặp kí tự này sẽ cắt nhỏ mảng

Char *token = strtok (array, "-");

dấu nhận biết

+ buffer buffer [33]

bd viết + itoa

itoa (i, buffer, 10)

chuyển i vào buffer hệ thập phân

* System ("code")

về ôn console

21/7/2022 * Quan trọng

thư viện <vector>

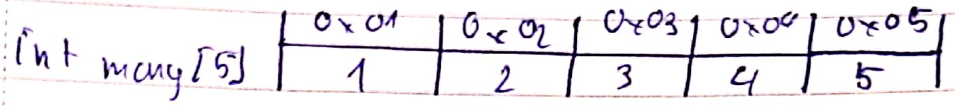
Linked list

Program counter

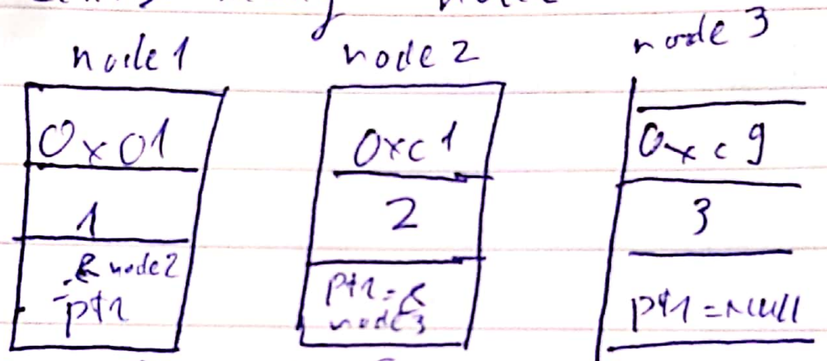
using namespace std;
vector <int> many = {1, 2, 3, 4} (C++)

in C:

node



vector <int> many node

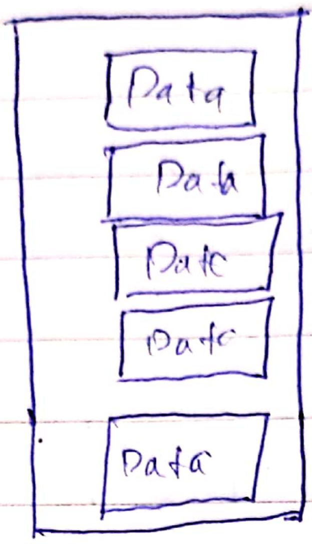


thêm ptr → lãng phí → ....

```
struct Linkelist
{
    int data;
    struct Linkedlist *next;      → con trỏ tới node tiếp theo
};
```

22/7/2022

Stack



Stack

Push: Thêm p.tử $\boxed{\text{data}}$ vào đỉnh, số p.tử +1

Pop: xoá p.tử đầu tiên, số p.tử -1

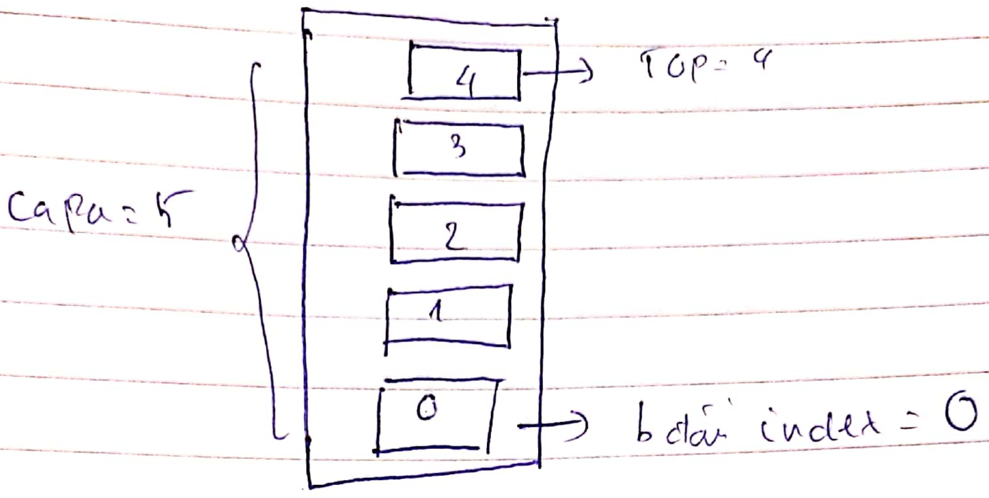
Top: Lấy g.trị của p.tử đ tiên

IsEmpty: K.tra ngăn xếp trống hay k?

IsFull: K.tra ngăn xếp đã đầy hay chưa

Size: Lấy số lg p.tử stack đang có

23/7/2022



khi ss lấy ss top ≥ Capa − 1

Push: ++top kc vì bđ = 0

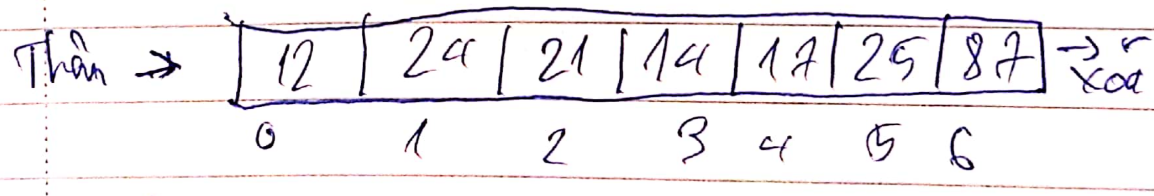
'\0' = NULL

∀ KDL: Queue kỹ thuật first in - first out

DeQueue lấy ptử đầu

EnQueue thêm ptử đầu vào

Front : lấy gtrị



front = 0

Rear = 6

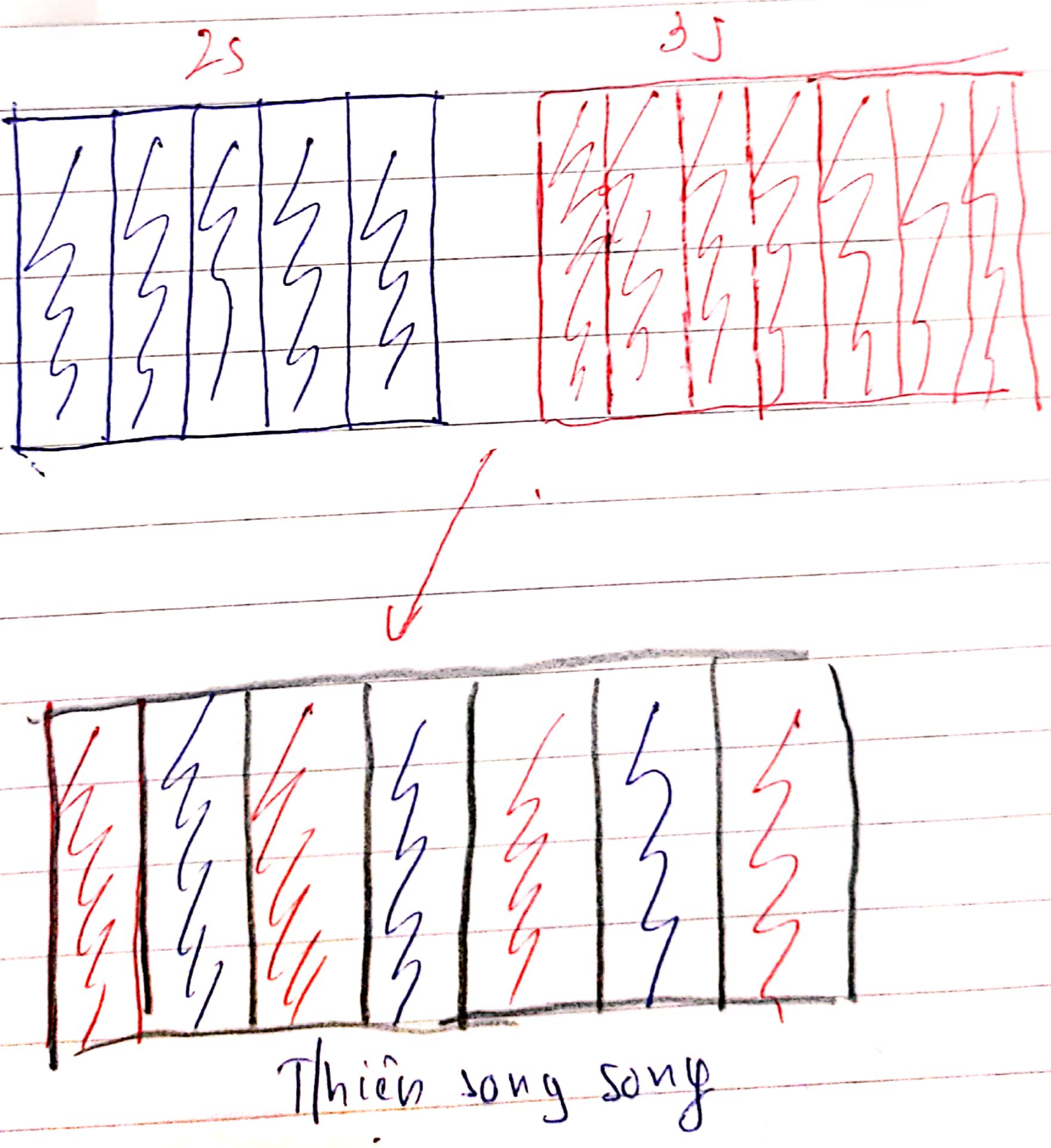
Dequeue
mang [0] = '\0';
front++

26/7/2022

Đa luồng trong C
Thread

Muốn chạy // Task 1 và task 2
ứng dụng RTOS

→ thư viện <pthread.h>



Thực thi song song