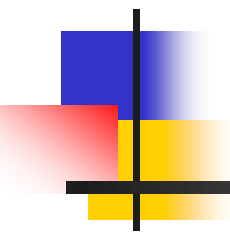


C Programming (CSE2035) (Chap12. Stacks)

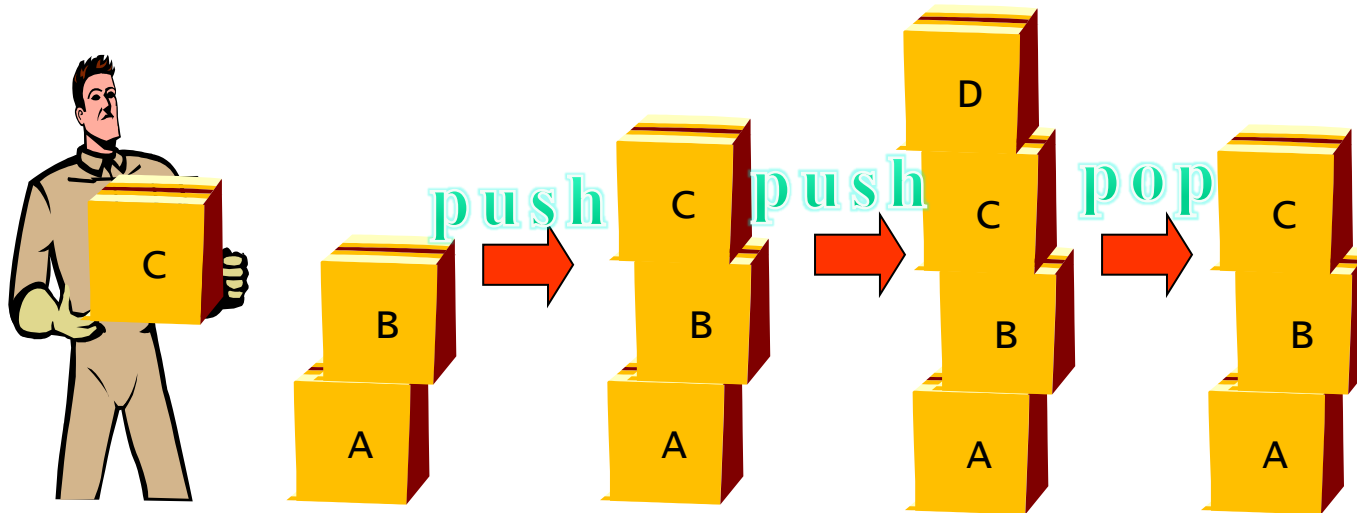


Sungwon Jung, Ph.D.

Bigdata Processing & DB LAB
Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea
Tel: +82-2-705-8930
Email : jungsung@sogang.ac.kr

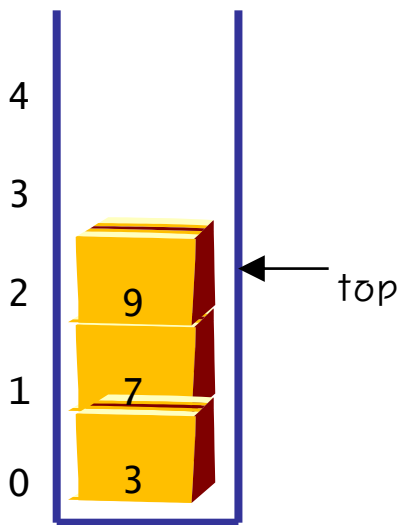
Stacks

- Stack은 linear list에서 모든 삽입과 삭제가 한 쪽 끝(Top)에서만 이루어지는 자료의 구조이다.
- LIFO(last in-first out): 가장 최근에 들어온 데이터가 가장 먼저 나감

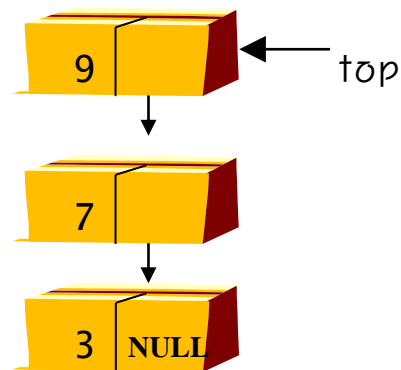


Linked Stack

- Linked list를 이용하여 구현한 stack
 - 장점: 크기가 제한되지 않음
 - 단점: 구현이 복잡하고 삽입이나 삭제 시간이 오래 걸린다.



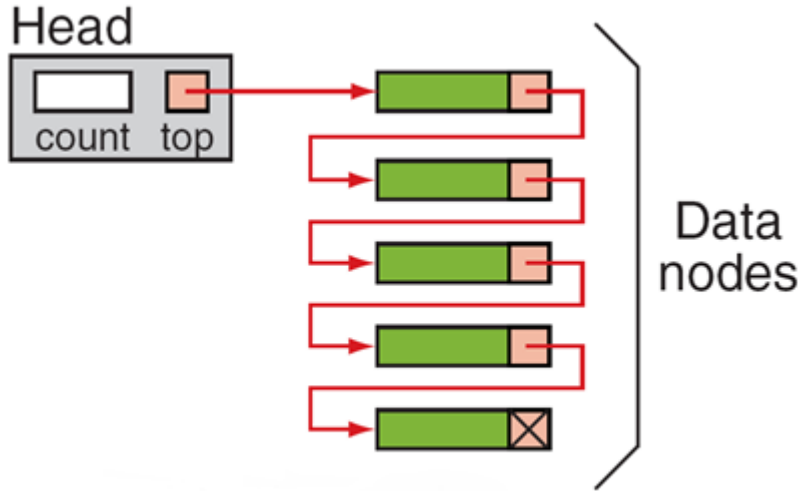
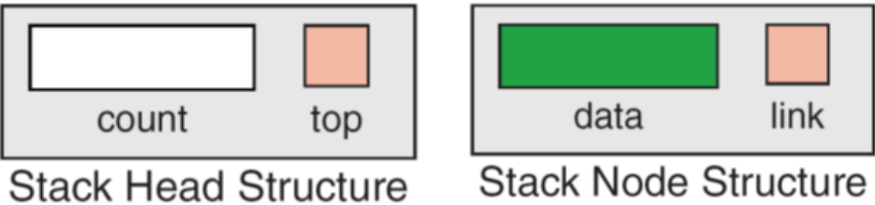
Array를 이용한 구현



Linked list를 이용한 구현

Linked Stack

- Linked list를 이용한 stack의 자료구조



```
typedef struct
{
    int count;
    struct node* top;
} STACK;

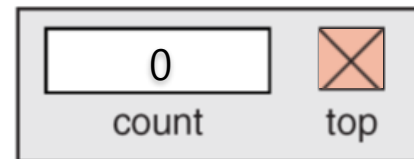
typedef struct node
{
    int data;
    struct node* link;
} STACK_NODE;
```

Linked Stack

- Linked list를 이용한 stack의 자료구조

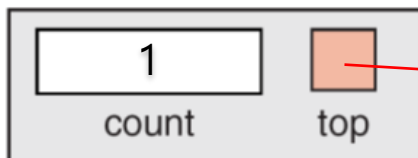
Before(init)

```
STACK* pStack;  
pStack = (STACK*)malloc(sizeof(STACK));  
pStack->top = NULL;  
pStack->count = 0;
```



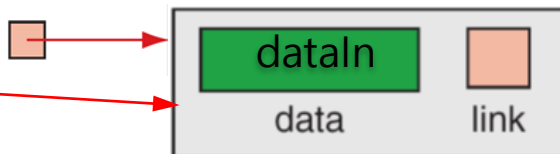
Stack Head Structure

After a single node is pushed



Stack Head Structure

pNew

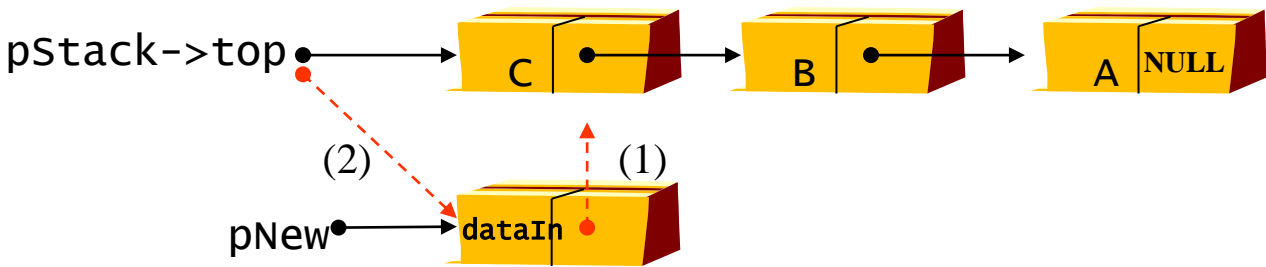


Stack Node Structure

```
pNew->data = dataIn;  
pNew->link = pStack->top;  
pStack->top = pNew;  
pStack->count++;
```

Push Stack

Stack의 push 연산



새로운 node에 공간을 할당

Stack의 top에 node를 push

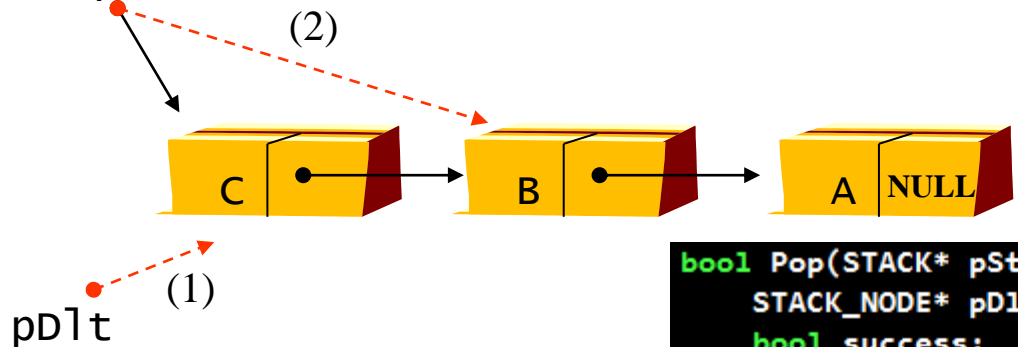
```
bool Push(STACK* pStack, int dataIn) {
    STACK_NODE* pNew;
    bool success;

    pNew = (STACK_NODE*)malloc(sizeof(STACK_NODE));
    if (!pNew) {
        success = false;
    } else {
        pNew->data = dataIn;
        pNew->link = pStack->top; (1)
        pStack->top = pNew; (2)
        pStack->count++;
        success = true;
    }
    return success;
}
```

Pop Stack

- Stack의 pop 연산

pStack->top



Empty stack인지 확인

삭제된 node의 공간을 free

```
bool Pop(STACK* pStack, int* dataOut){
    STACK_NODE* pDlt;
    bool success;

    if(pStack->top) {
        success = true;
        *dataOut = pStack->top->data;
        (1) pDlt = pStack->top;
        (2) pStack->top = (pStack->top)->link;
        pStack->count--;
        free(pDlt);
    } else {
        success = false;
    }
    return success;
}
```