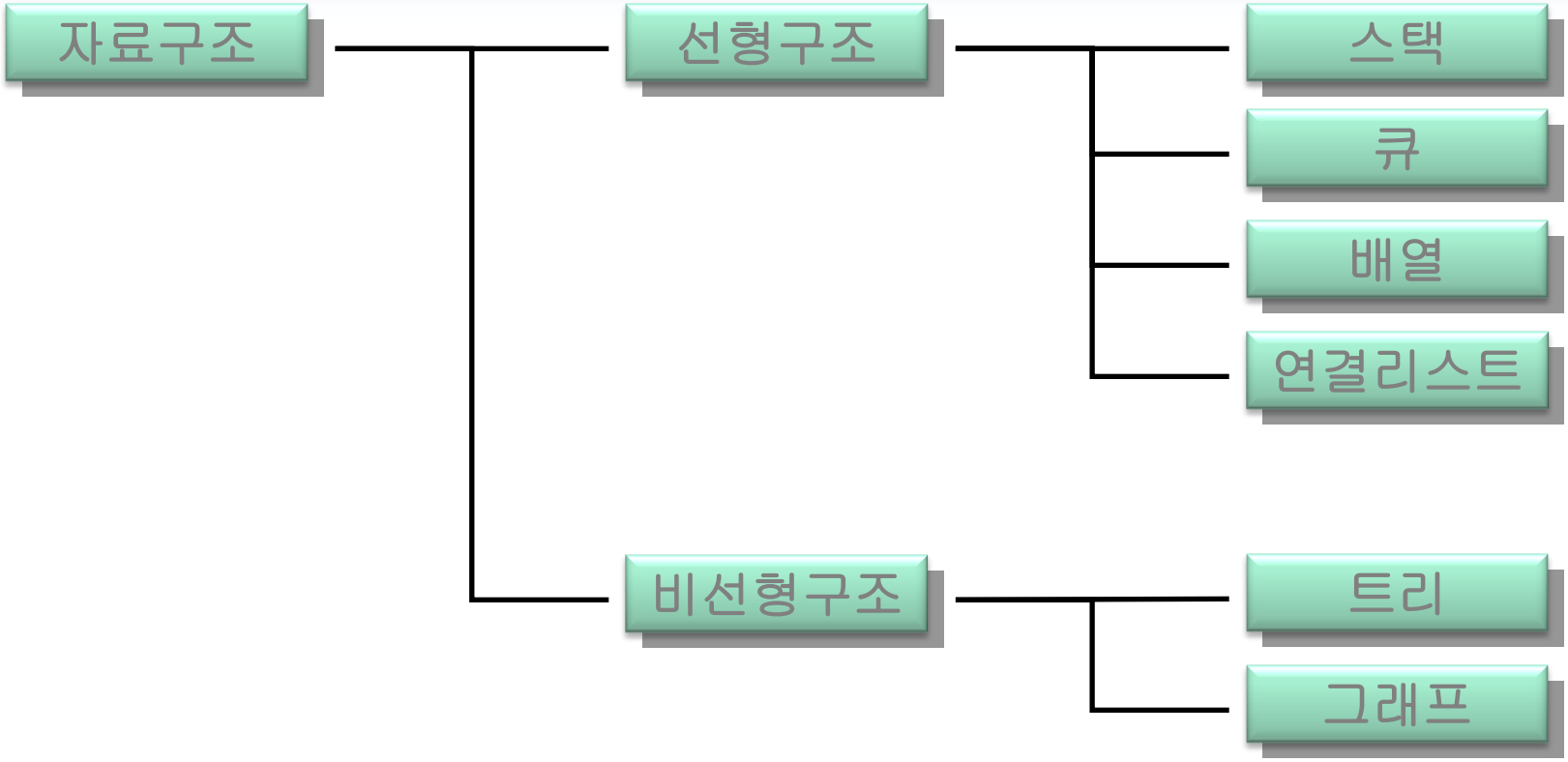
2강

DATA STRUCTURE

자료구조의 개념

- 자료 : 임의의 작업을 위하여 소용되는 것을 일컫는 말
- 자료구조 : 개개의 자료 항목끼리 상호 연관 관계를 갖는 것
- 선형구조
  - 데이터의 전후 항목 사이의 관계가 1:1
  - 선후관계가 명확한 한 줄기 선의 형태유지
  - 스택, 큐, 선형리스트, 연결리스트, 데크
- 비선형 구조
  - 데이터 항목 사이의 관계가 1:n, n:m
  - 트리, 그래프

자료구조의 관계



선형구조

- **스택**
  - LIFO(Last Input First Output)
  - 스택포인트(Top)로 삽입, 삭제 처리
- **큐**
  - FIFO(First Input First Output)
  - Front, Rear로 삽입, 삭제 처리
- **배열**
  - 같은 형태의 요소들로 구성
- **연결리스트**
  - 실질적인 위치에 대한 정보를 갖는 포인터에 의해 선형구조 형성

비선형 구조

❖ 트리
- 계층적 구조, 레벨 개념

❖ 그래프
- 정점과 연결선의 집합으로 구성
- 정　점 : 데이터 표시
- 연결선 : 데이터들의 관계 표현

알고리즘

❖ 어떤 정해진 처리를 위한 명령들의 순서 집합

❖ 알고리즘의 조건

- 입력 : 외부로부터 제공되는 자료가 입력되어야 한다.
- 출력 : 적어도 하나 이상의 결과를 생성한다.
- 명확성 : 각 명령들이 모호하지 않고 명확하다.
- 유한성 : 어떤 경우에도 한정된 단계를 실행한 후에는 끝낸다.
- 실제성 : 알고리즘의 모든 명령이 실행 가능하다.

최적 알고리즘

- **시간 복잡도**
  - 어떤 문제를 처리하기 위해 특정 알고리즘이 수행되는 기본 연산수
- **공간 복잡도**
  - 필요로 하는 메모리의 크기

스택

❖ 정의

- 영어로 건초, 쌓아올린 건초더미 지칭
- 후입선출, LIFO(Last Input First Output)

❖ 용도

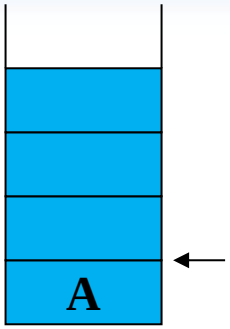
- 수식연산, 부프로그램 호출시 복귀주소 관리

❖ 특징

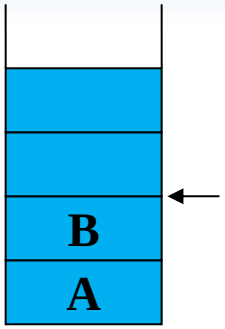
- Top이라는 스택포인터를 사용하여 가장 나중에 입력된 데이터의 위치 지정

Stack 의 입출력

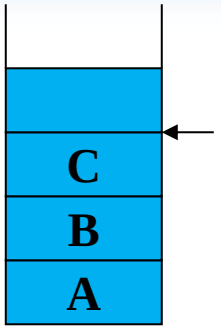
(1) 초기상태
Top : 0



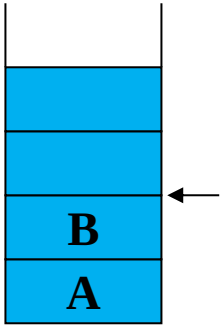
(2) A입력 후
Top : 1



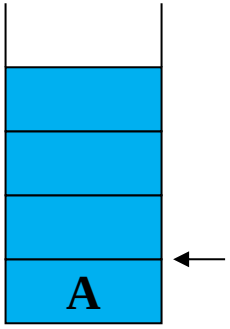
(3) B입력 후
Top : 2



(4) C입력 후
Top : 3



(5) C출력 후
Top : 2



(6) B출력 후
Top : 1

struct / get_node()

```
#include <stdio.h>
#include <stdlib.h>
#define EMPTY 0

struct node {
        int data;
        struct node * link;
};
typedef struct node Stack;

Stack * GetNode(){
        Stack * tmp;
        tmp=(Stack *)malloc(sizeof(Stack));
        tmp->link=EMPTY;
        return tmp;
}
```

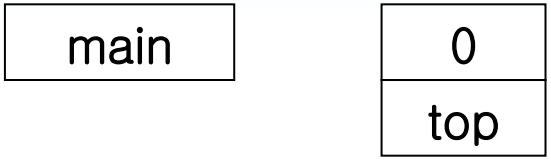
Push()

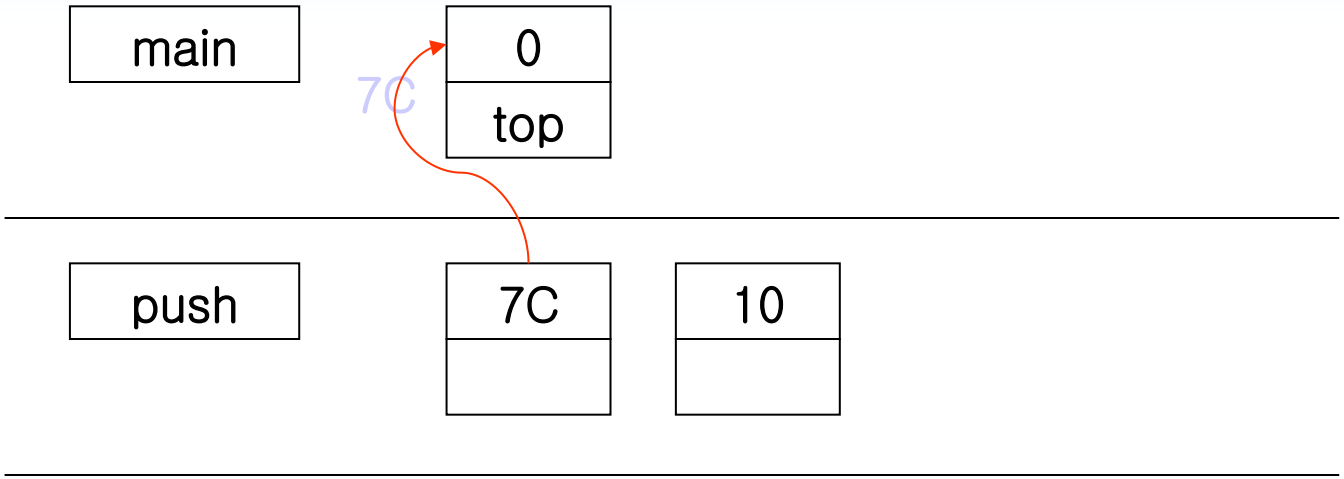
```
void Push(Stack **top, int data){
        Stack *tmp;
        tmp=*top;

        *top=GetNode();

        (*top)->data=data;
        (*top)->link=tmp;
}
```

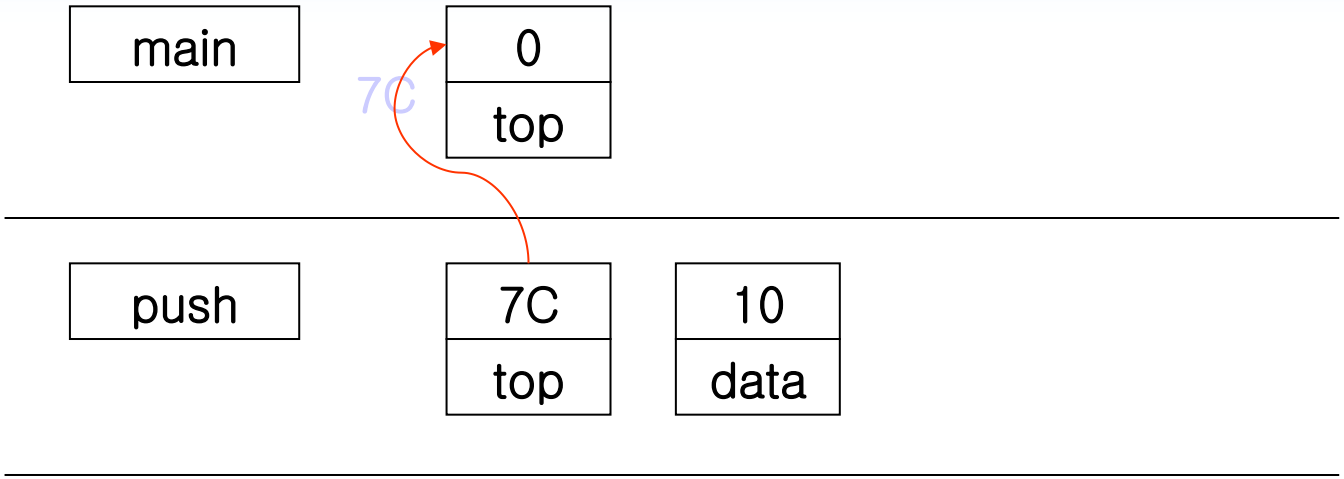
main()

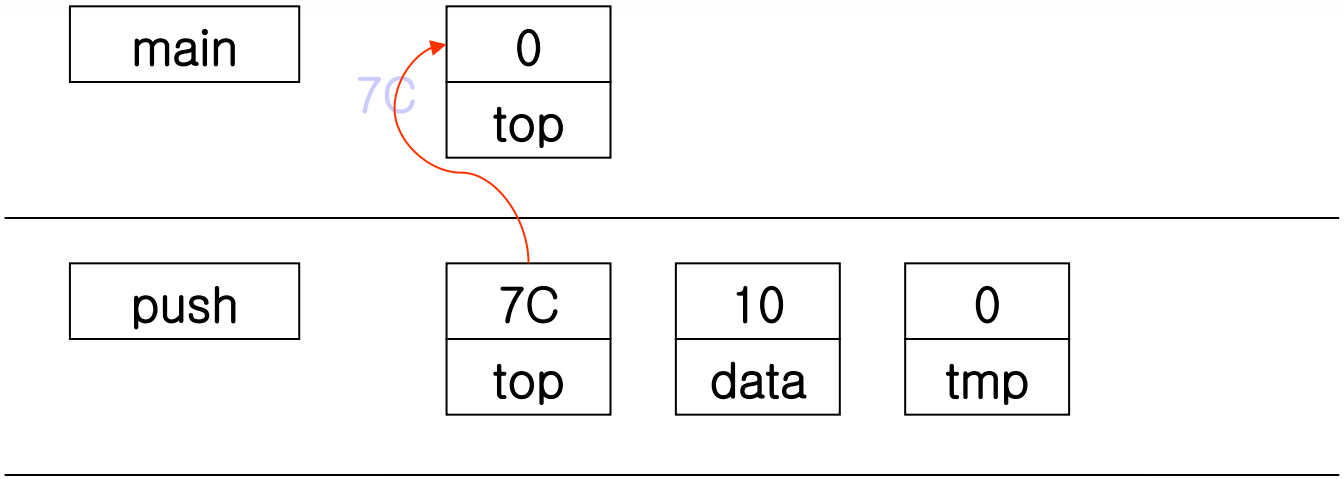
```
void main () {
        Stack * top=EMPTY;

        Push(&top, 10);
        Push(&top, 20);
        Push(&top, 30);

        printf("%d", Pop(&top));
        printf("%d", Pop(&top));
        printf("%d", Pop(&top));
}
```

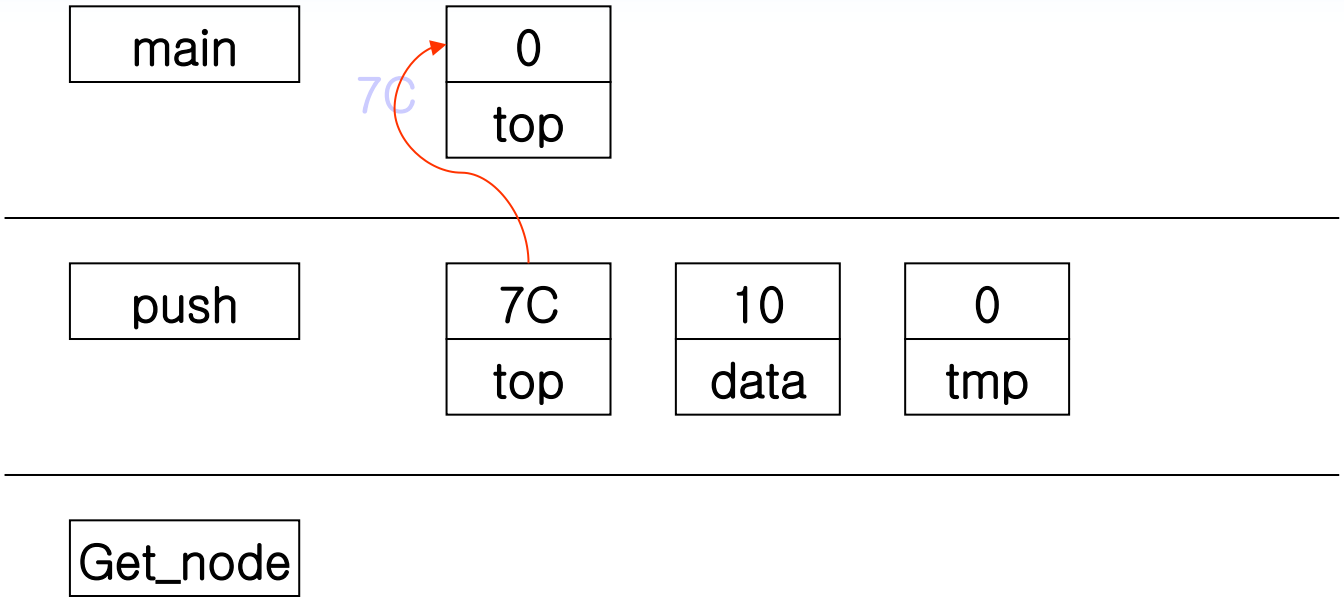
push

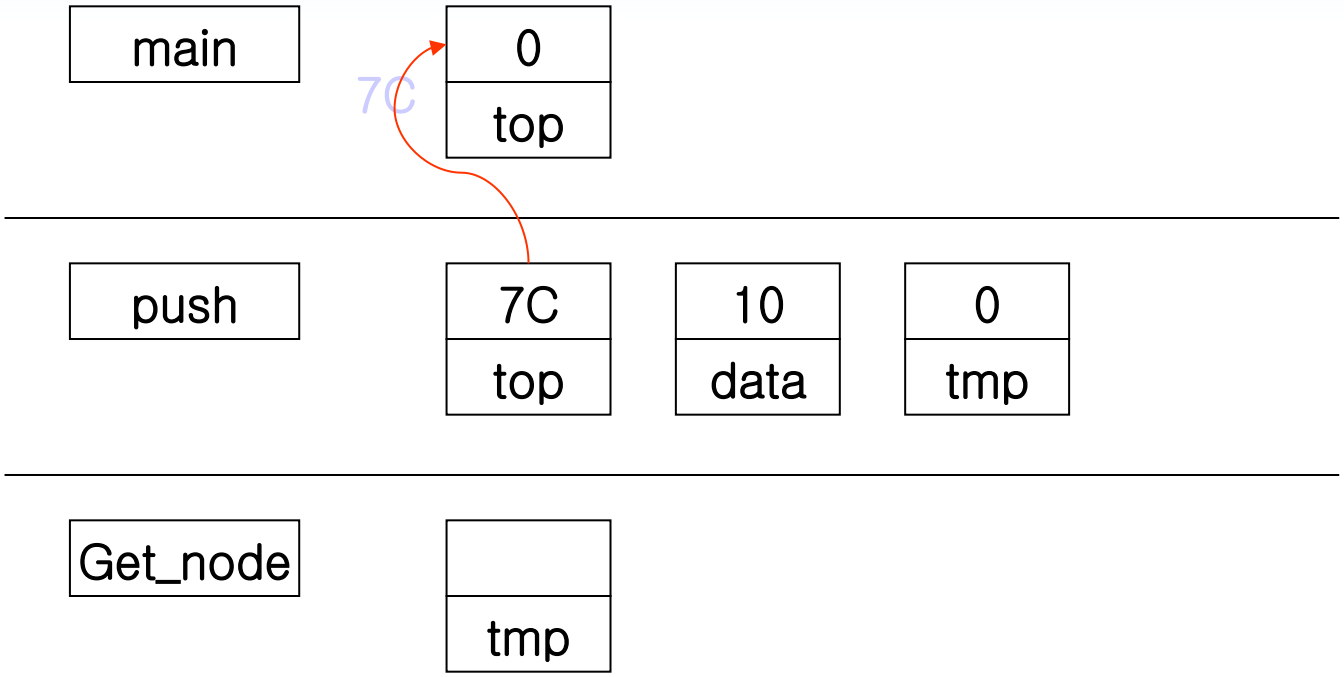


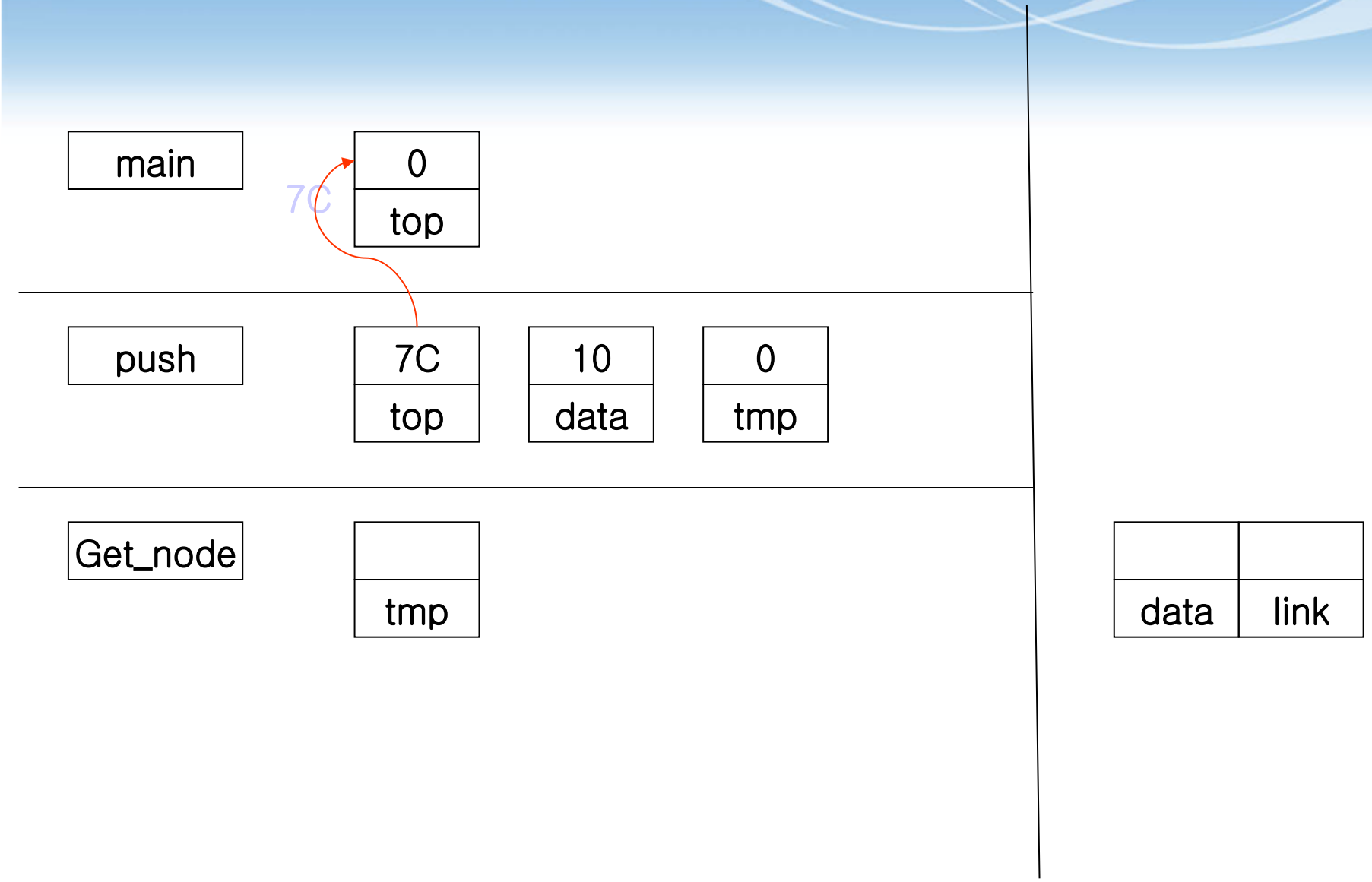

main
0
top
7C
push
7C
10

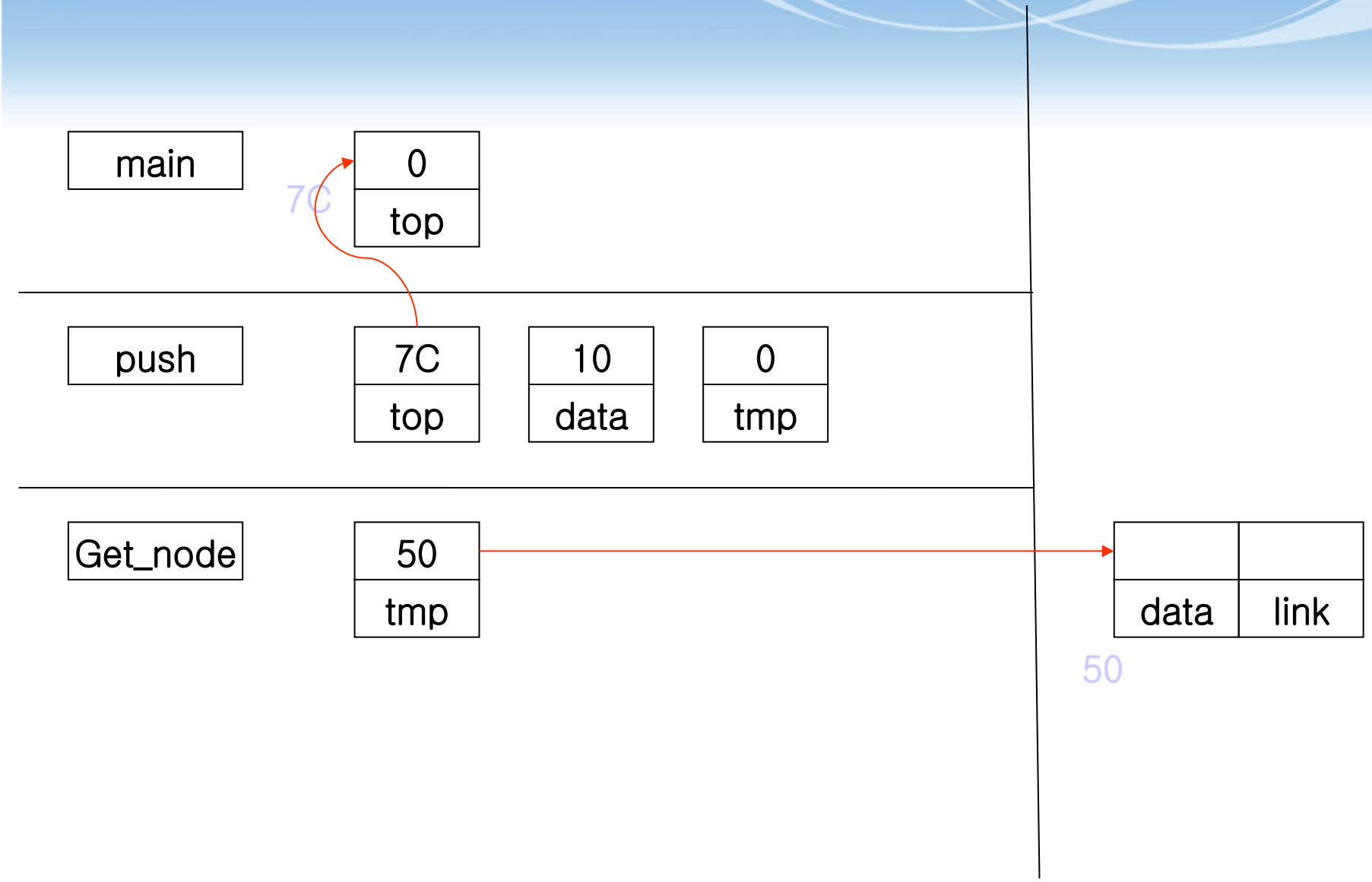

main
0
top
7C
push
7C
top
10
data

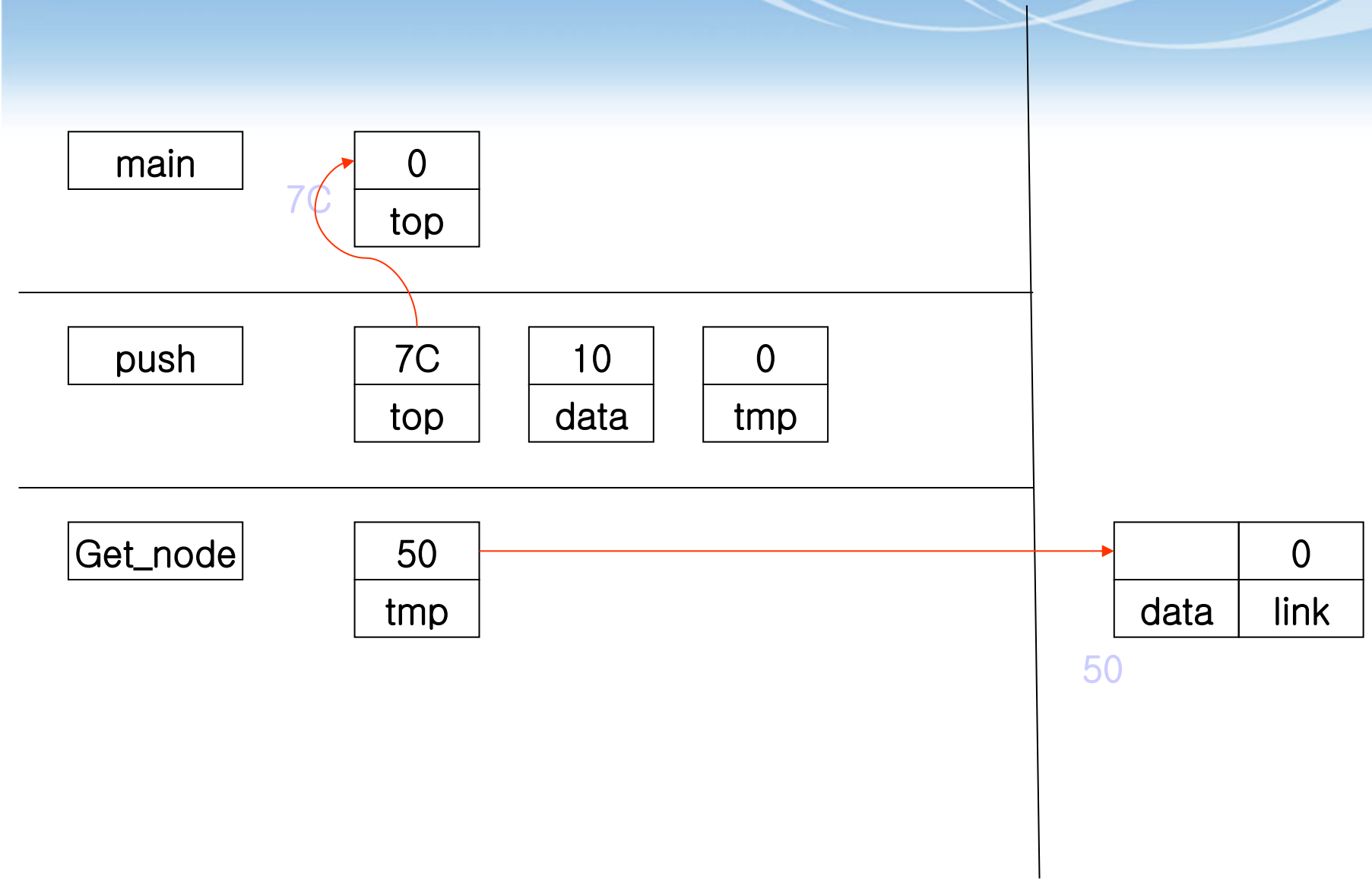

main
0
top
7C
push
7C
top
10
data
0
tmp

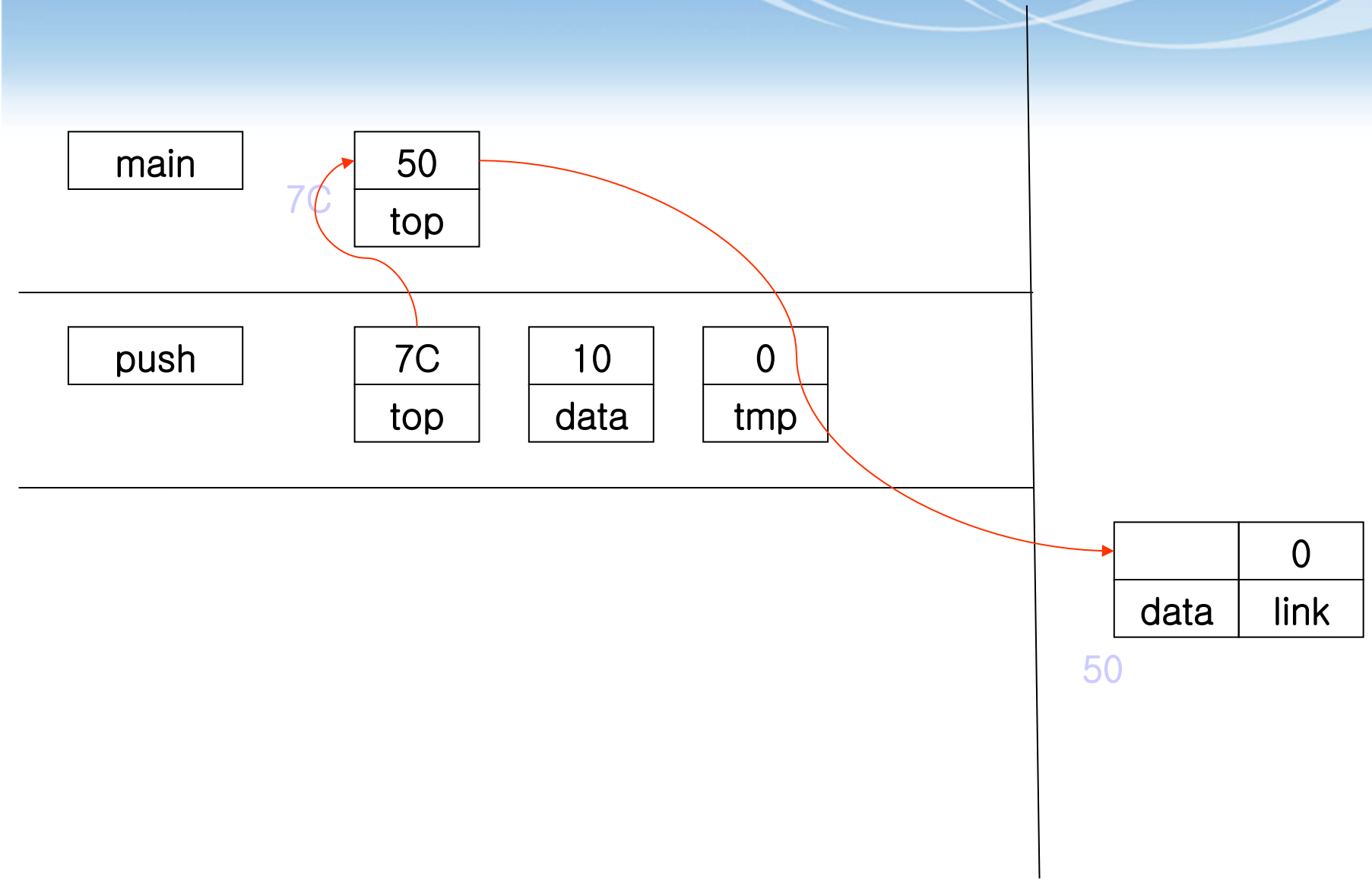

main
0
top
7C
push
7C
top
10
data
0
tmp
Get_node

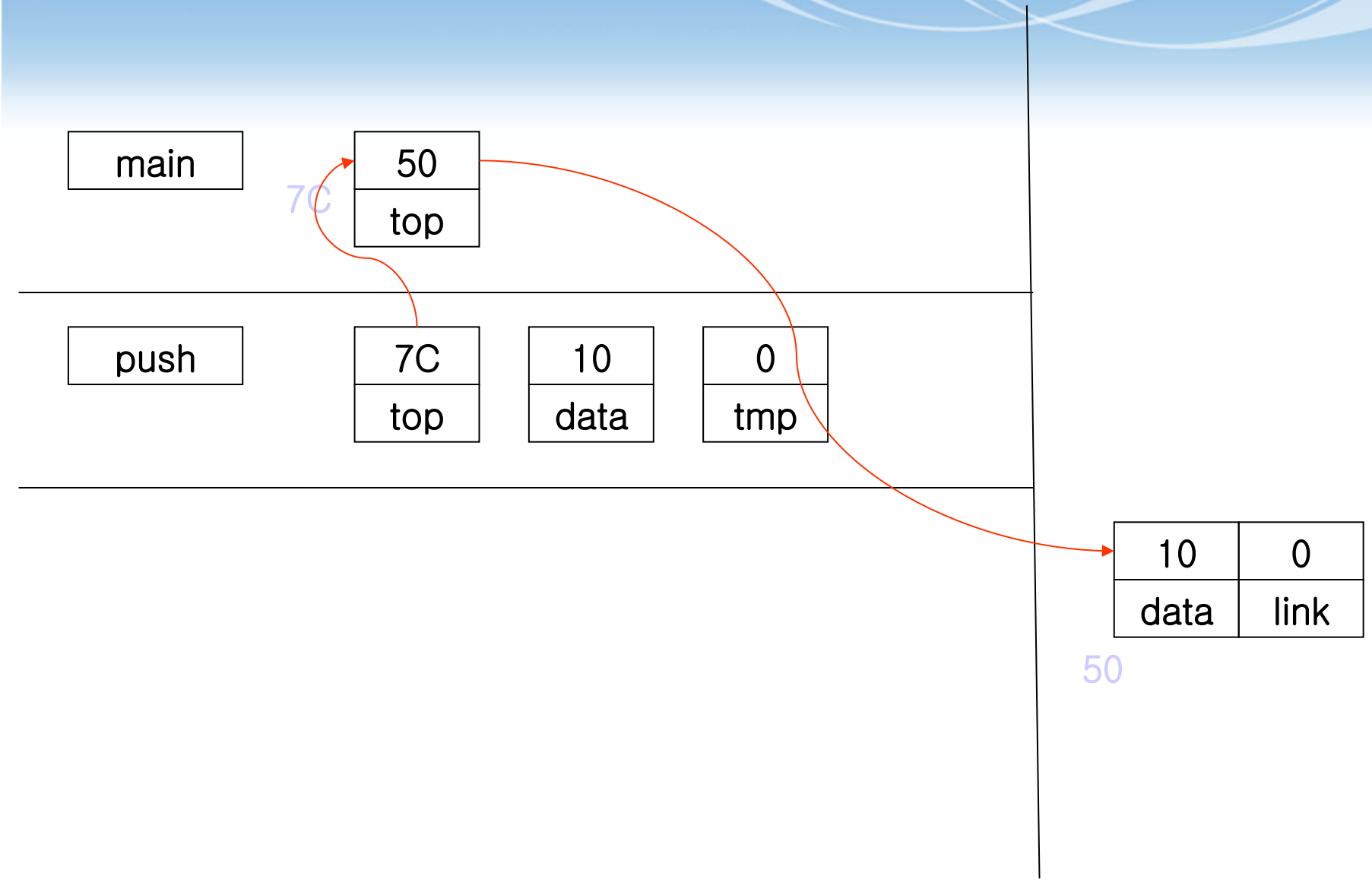

main
0
top
7C
push
7C
top
10
data
0
tmp
Get_node
tmp

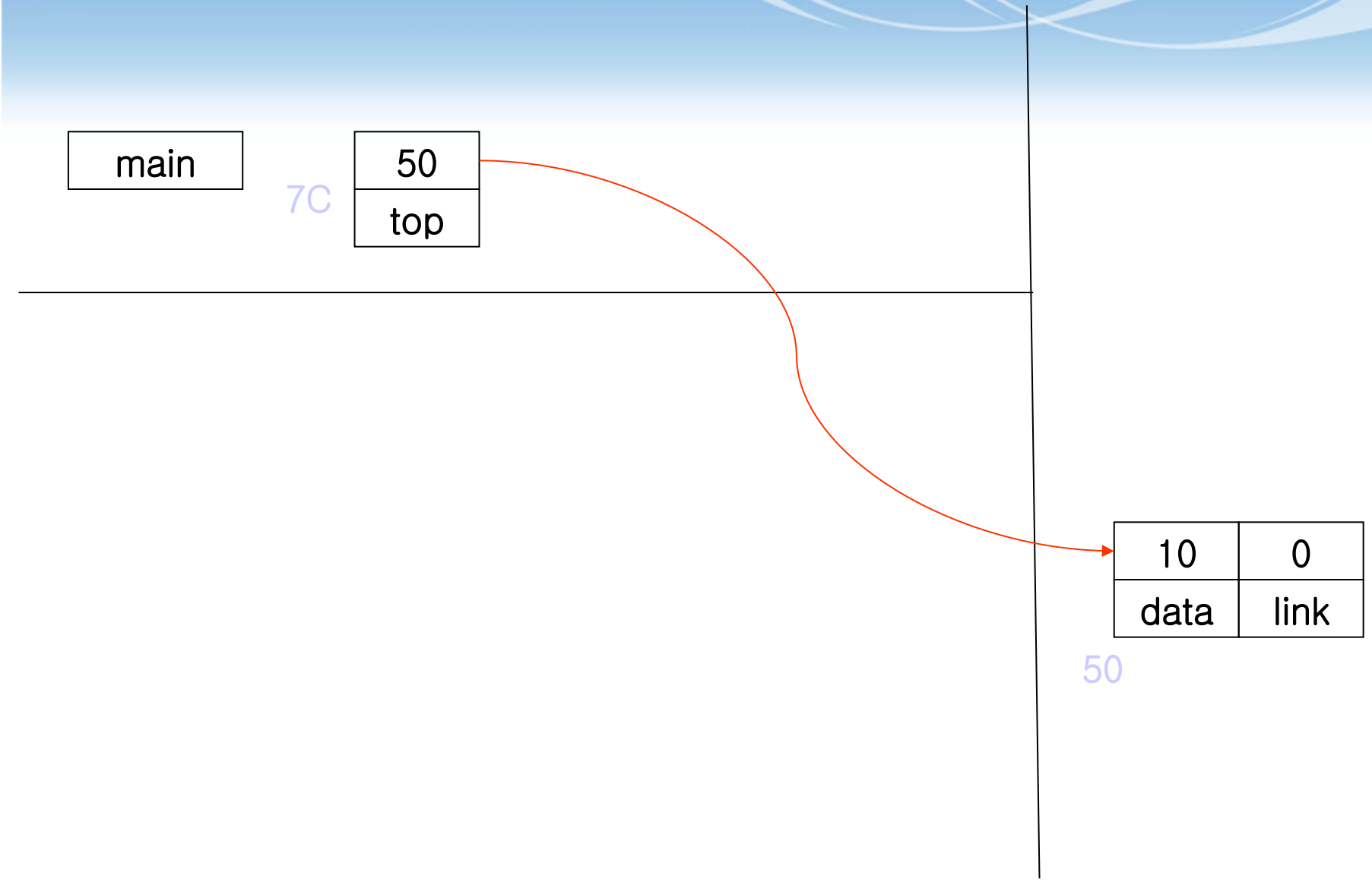

main
0
top
7C
push
7C
top
10
data
0
tmp
Get_node
tmp
data
link

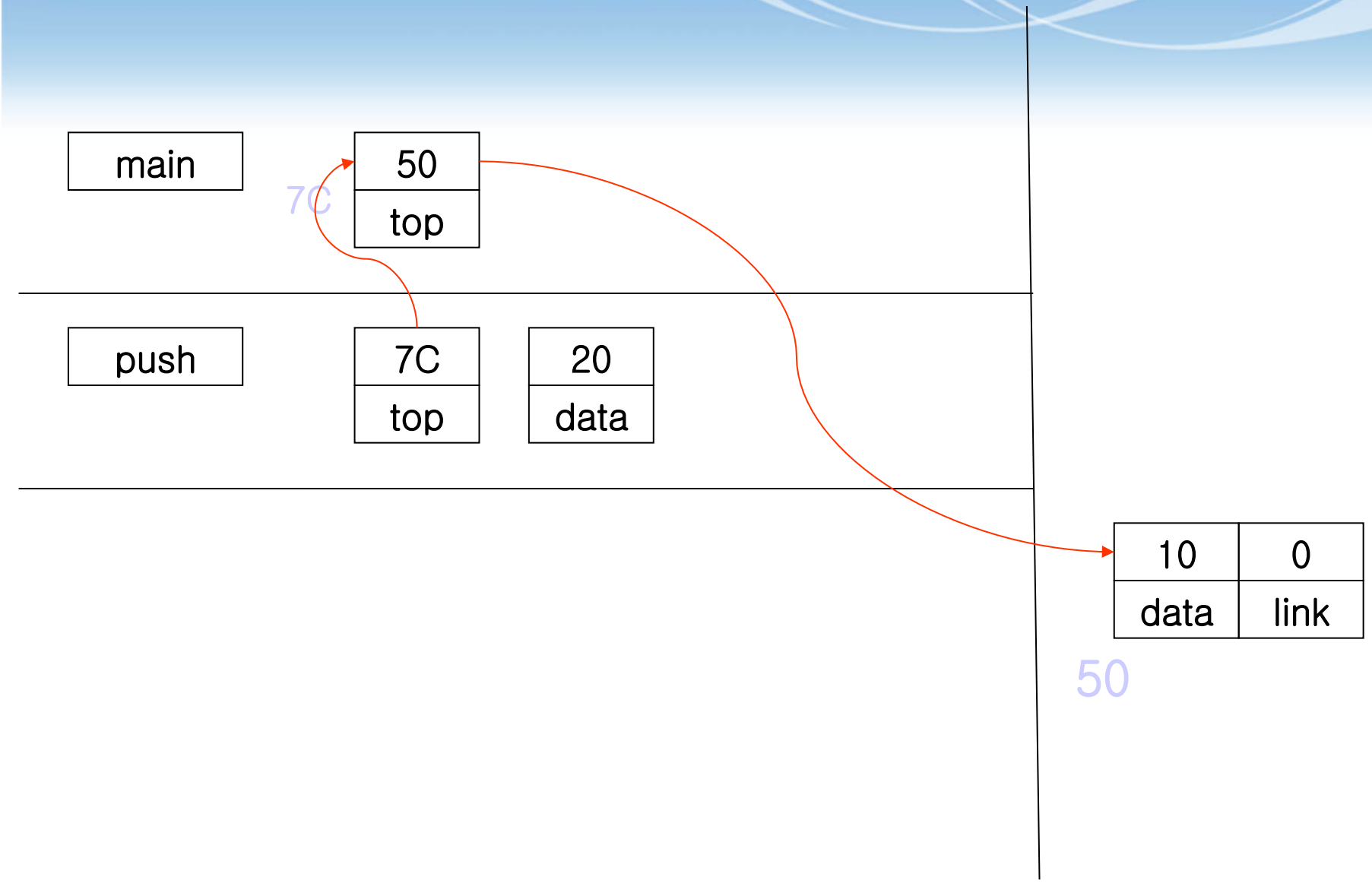

main
0
top
7C
push
7C
top
10
data
0
tmp
Get_node
50
tmp
data
link
50

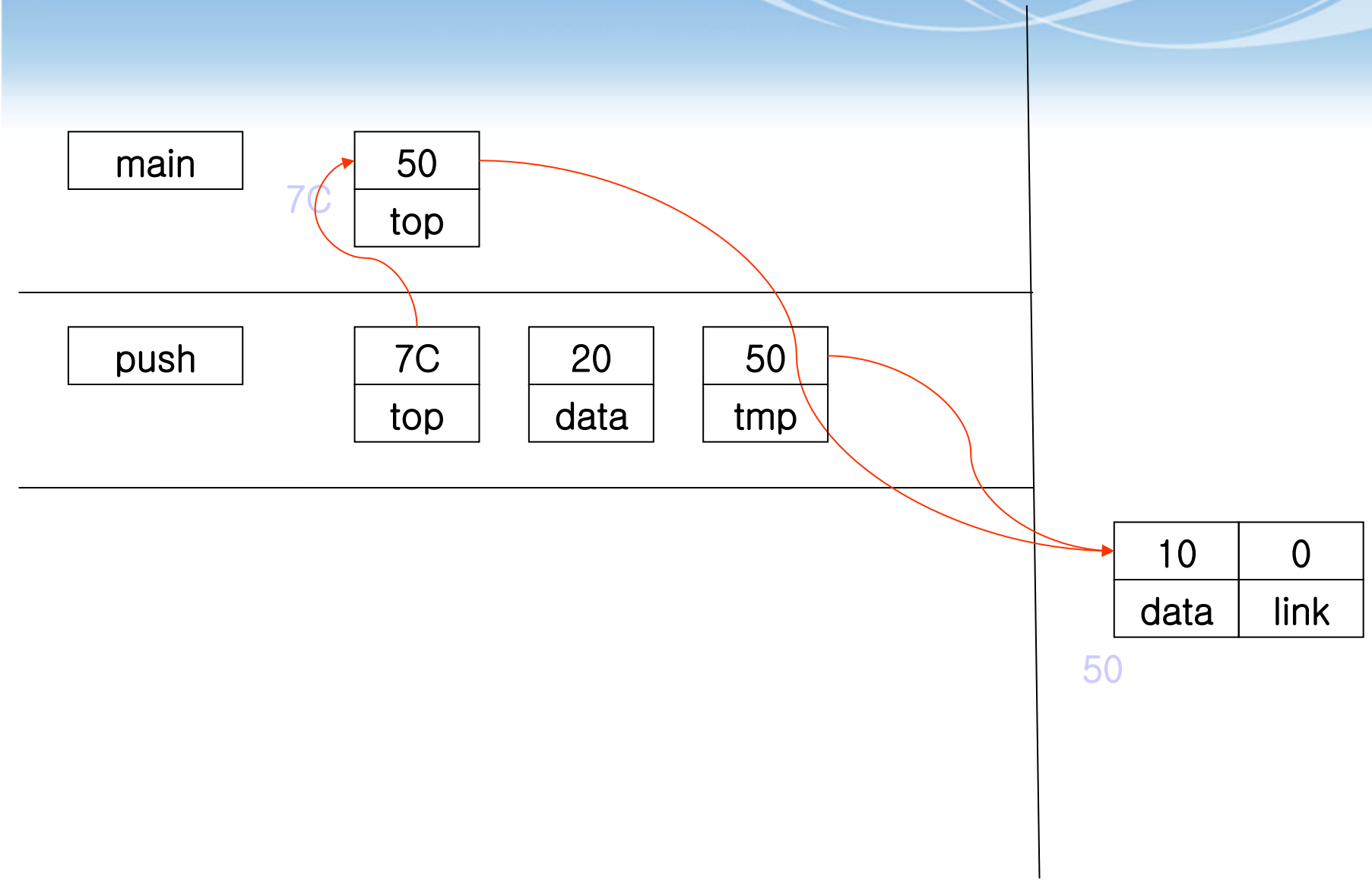

main
0
top
7C
push
7C
top
10
data
0
tmp
Get_node
50
tmp
data
link
0
50

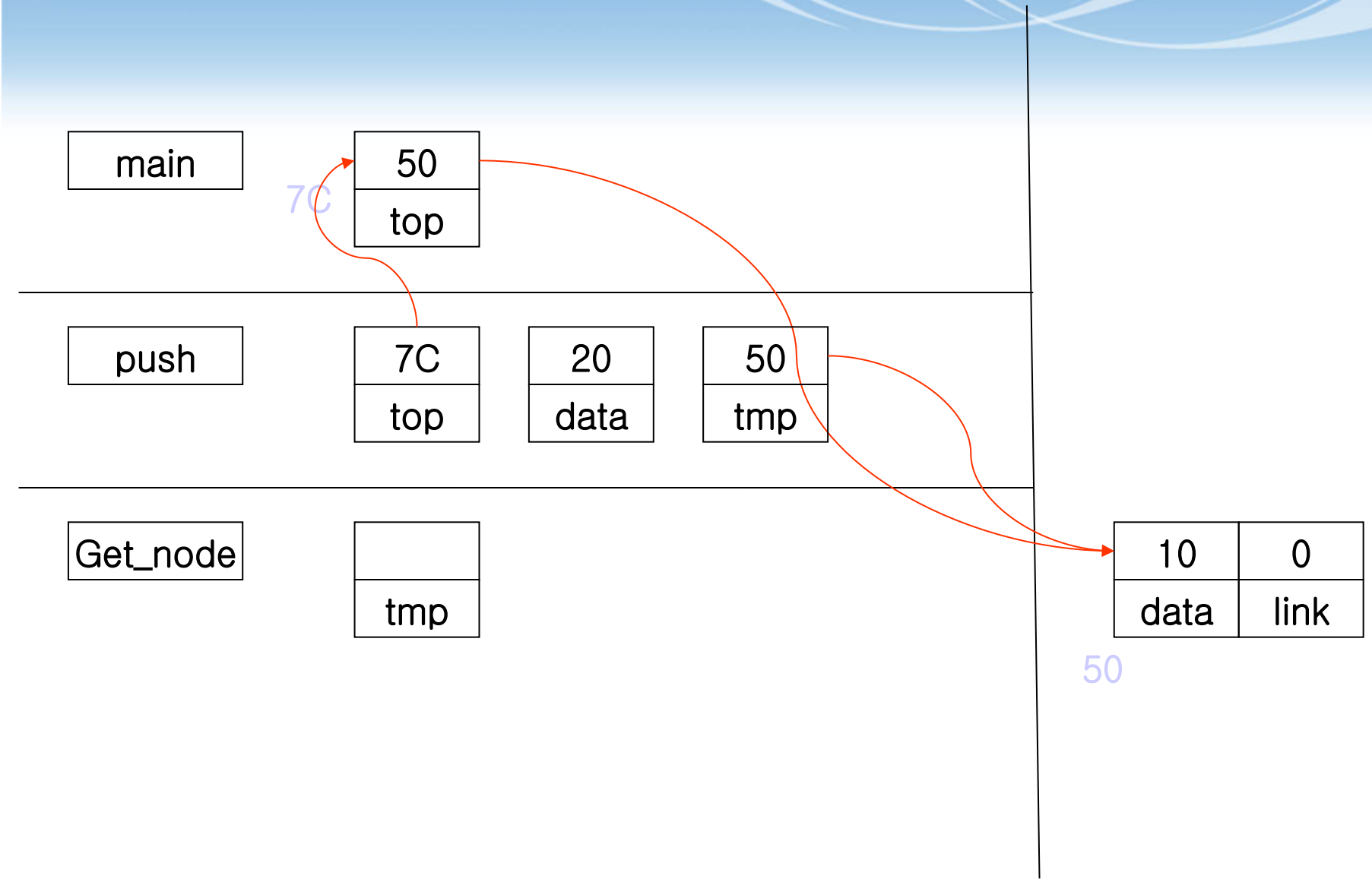

main
50
top
7C
push
7C
top
10
data
0
tmp
0
data
link
50

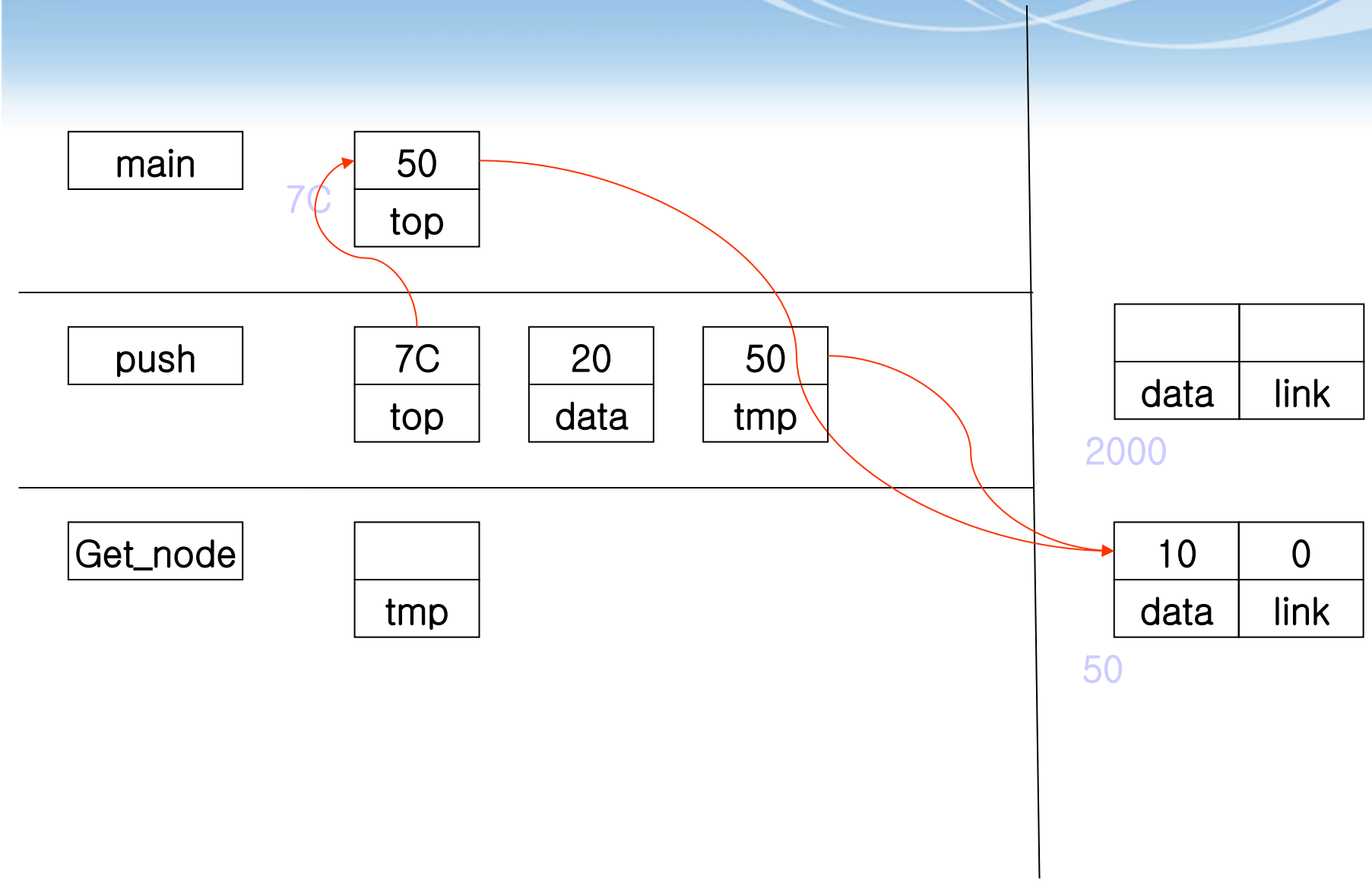

main
50
top
7C
push
7C
top
10
data
0
tmp
10
data
0
link
50

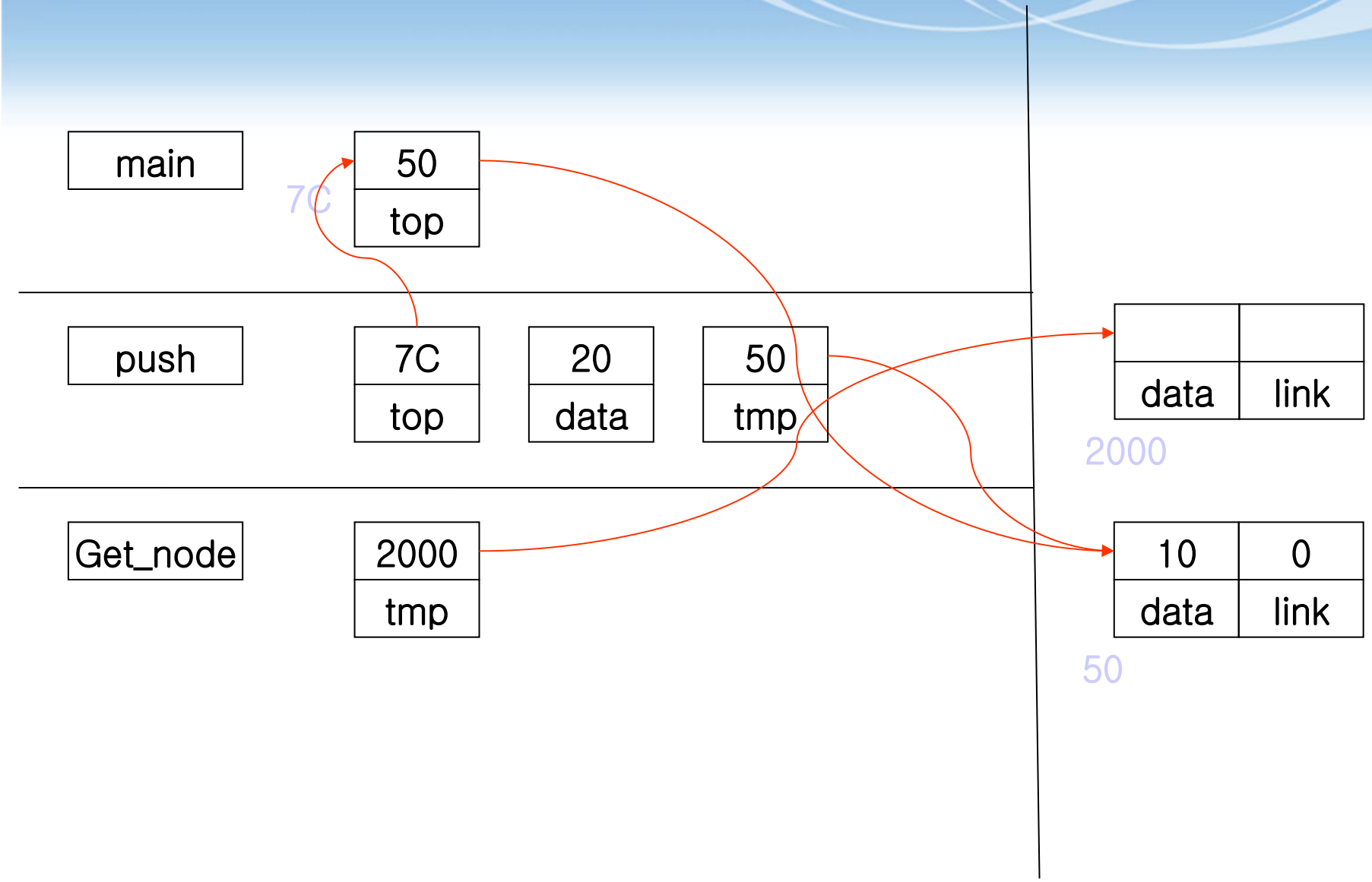

main
50
top
7C
10
0
data
link
50

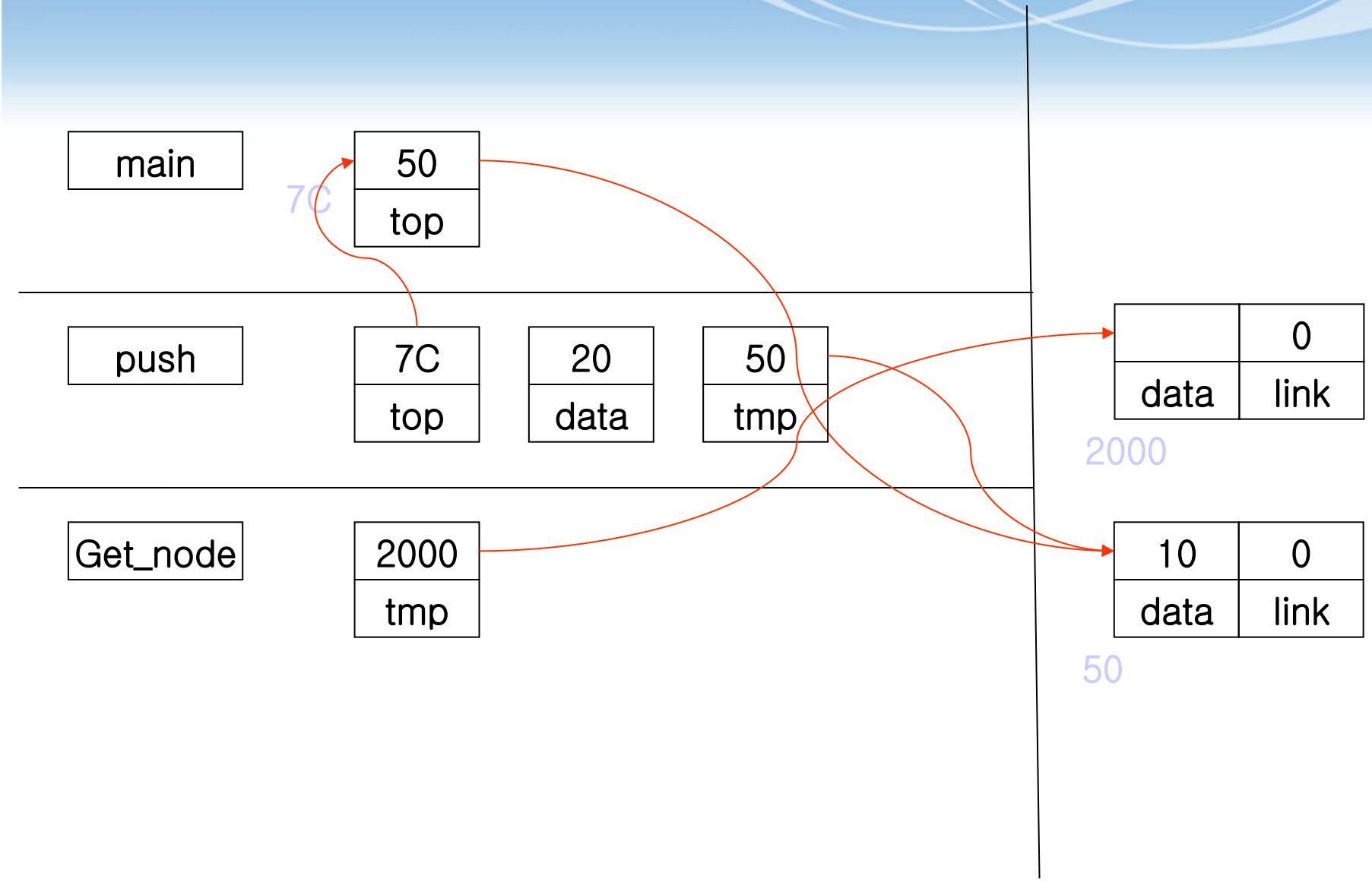

main
50
top
7C
push
7C
top
20
data
10
data
0
link
50

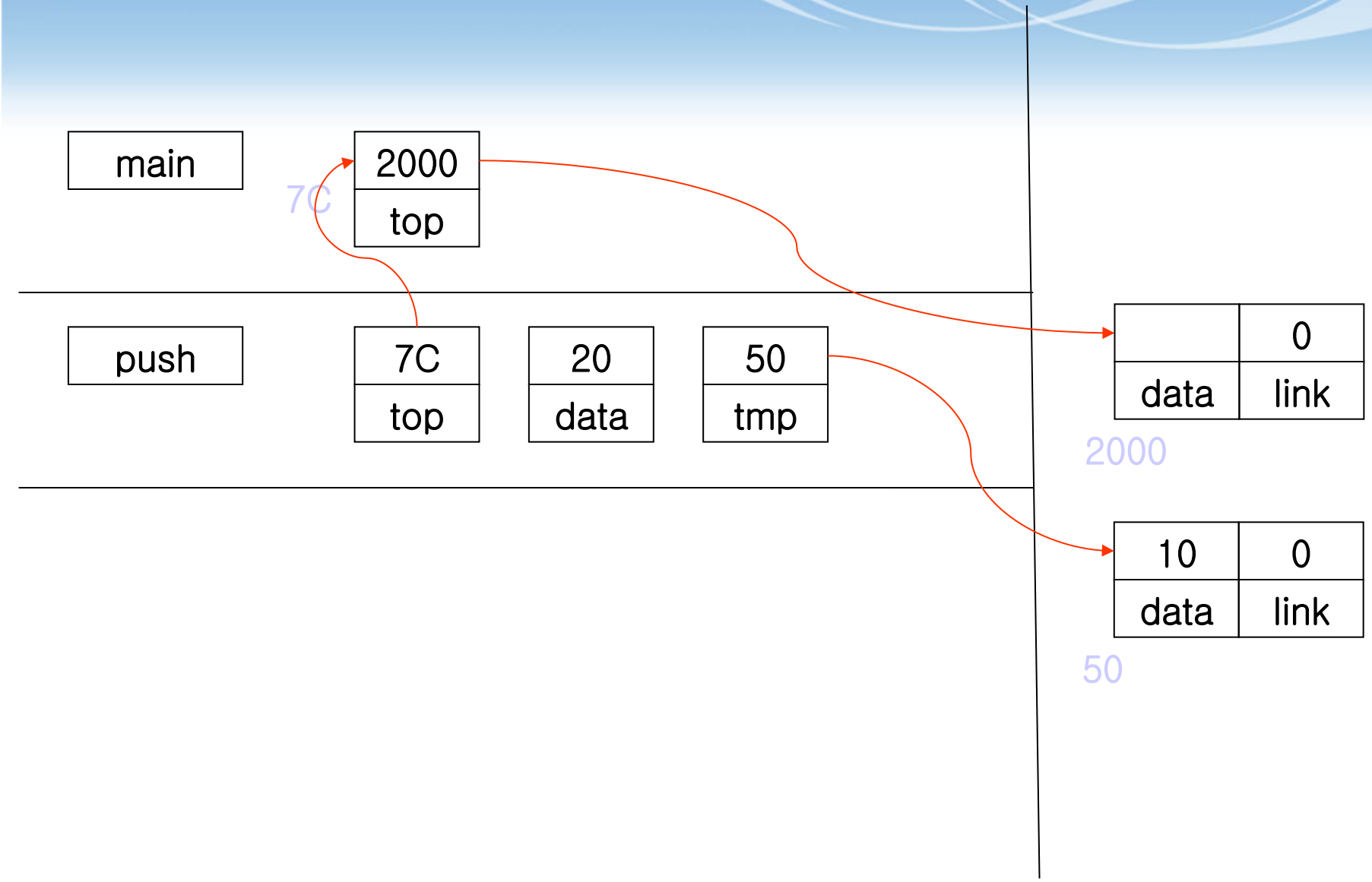

main
50
top
7C
push
7C
top
20
data
50
tmp
10
data
0
link
50

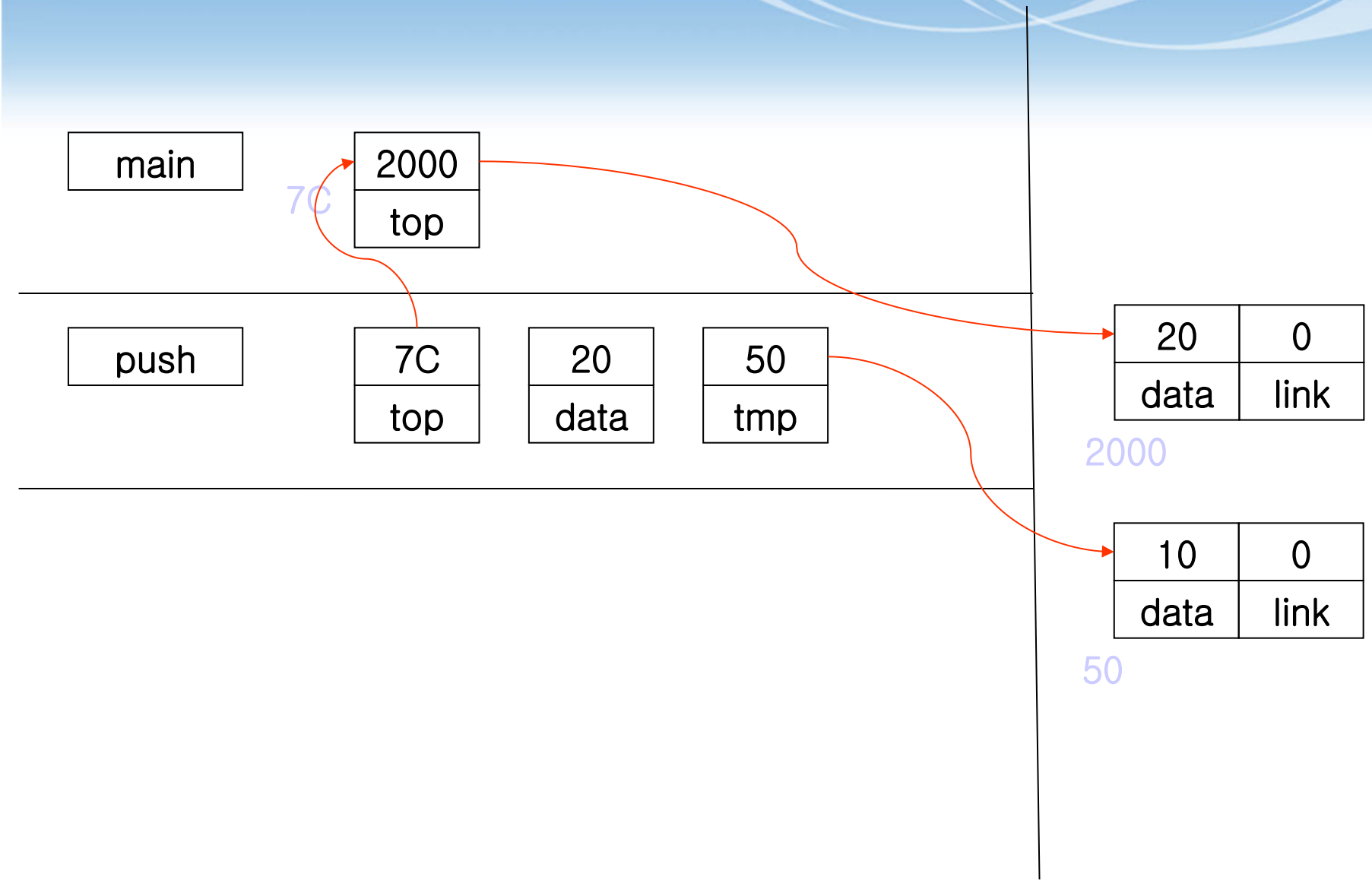

main
50
top
7C
push
7C
top
20
data
50
tmp
Get_node
tmp
10
data
0
link
50

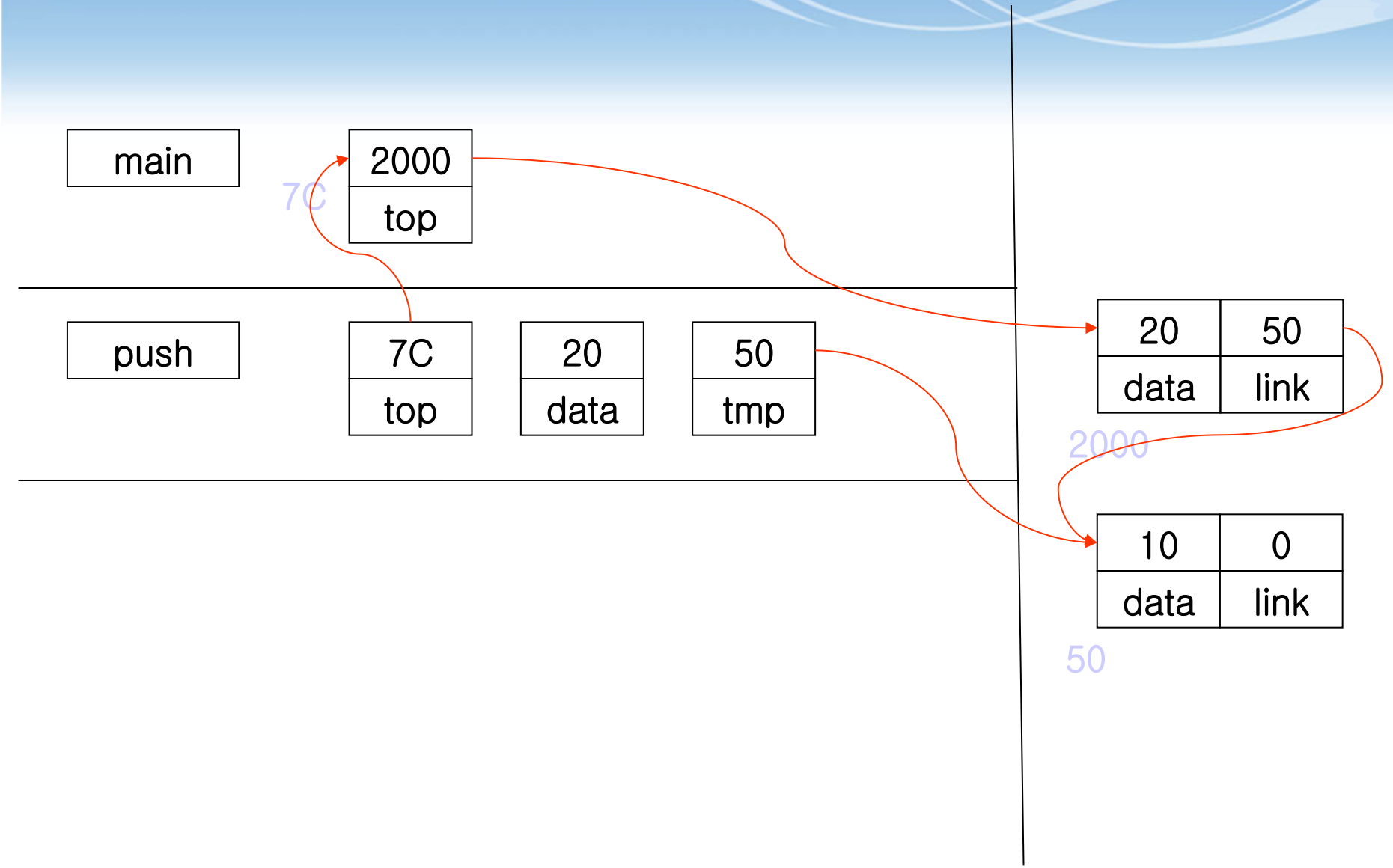

main
50
top
7C
push
7C
top
20
data
50
tmp
data
link
2000
Get_node
tmp
10
0
data
link
50

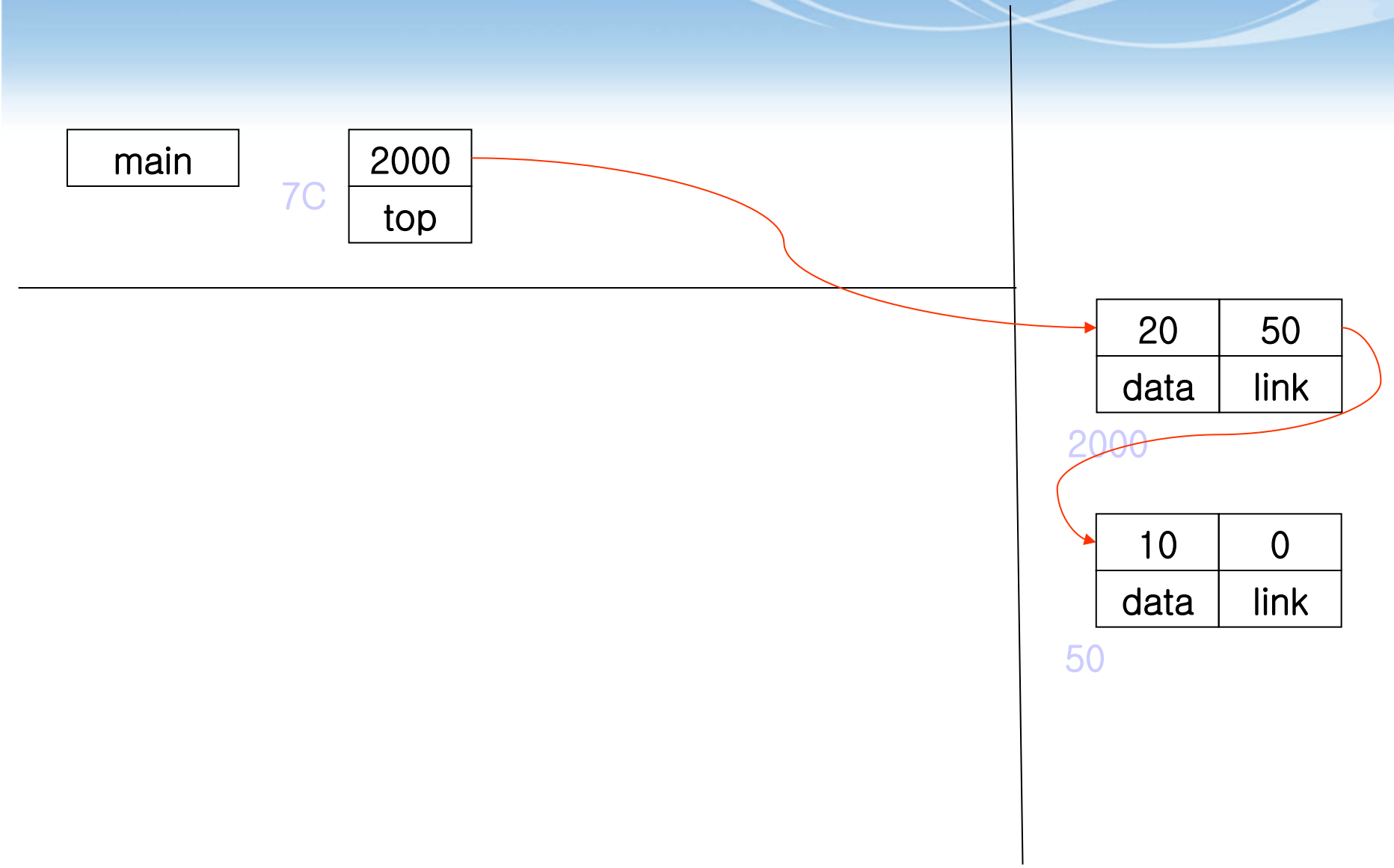

main
50
top
7C
push
7C
top
20
data
50
tmp
Get_node
2000
tmp
data
link
2000
10
0
data
link
50


main
50
top
7C
push
7C
top
20
data
50
tmp
Get_node
2000
tmp
0
data
link
2000
10
0
data
link
50


main
2000
top
7C
push
7C
top
20
data
50
tmp
0
data
link
2000
10
0
data
link
50


main
2000
top
7C
push
7C
top
20
data
50
tmp
20
0
data
link
2000
10
0
data
link
50


main
2000
top
7C
push
7C
top
20
data
50
tmp
20
50
data
link
2000
10
0
data
link
50


main
2000
top
7C
20
50
data
link
2000
10
0
data
link
50