Berkeley DB for TinyIoT

Sejong Univ.

Name : Park Minji
E-mail : iorw0224@gmail.com

Part 1

Sub와 다른 DB 구조상에 저장 방식의 차이

기존 저장 방식

Key	Value
aei	TAE1
aei	TAE3
aei	TAE2
api	tinyProject1
api	tinyProject3
api	tinyProject2
ct	20220513T083900
ct	20220513T083900
ct	20220513T083900
et	20240513T083900
et	20240513T083900
et	20240513T083900
lt	20220513T083900
lt	20220513T083900
lt	20220513T083900
pi	5-20191210093452845
pi	5-20191210093452845
pi	5-20191210093452845
ri	TAE1
ri	TAE3
ri	TAE2
rn	Sensor1
rn	Sensor3
rn	Sensor2
rr	true
rr	true
rr	true
ty	2
ty	2
ty	2

SUB 저장 방식

Key(label)	Value(uri)
3-20210406084023203796	23-20220406846532993O
3-20210406084023203796	Sub1
3-20210406084023203796	http://223.131.176.101:3000/ct=json
3-20210406084023203796	1 (net)
3-20210406084023203796	test/test/test/test1
3-20210406084023203796	20220406T084653
3-20210406084023203796	20260406T084653
3-20210406084023203796	20220406T084653
3-20210406084023203796	23
3-20210406084023203796	1
3-20210406084023203796	23-20210406846532993O
3-20210406084023203796	Sub2
3-20210406084023203796	http://223.131.176.101:3000/ct=json
3-20210406084023203796	4 (net)
3-20210406084023203796	test/test/test/test2
3-20210406084023203796	20210406T084653
3-20210406084023203796	20250406T084653
3-20210406084023203796	20210406T084653
3-20210406084023203796	23
3-20210406084023203796	1

저장 구조 변경이 힘든 이유

SUB 구조체의 key로 들어오는 pi값은 모두 같아서 DB에 들어온 순서를 명확하게 할 수 있었지만,
SUB가 아닌 다른 구조체의 경우 ri를 key값으로 설정하는 경우 들어온 순서가 아닌 길이순, 사전순으로 저장되기 때문에 들어온 순서를 명확하게 하기 어려움

Part 1

Sub와 다른 DB 구조상에 저장 방식의 차이

기존 저장 방식

Key	Value
aei	TAE1
aei	TAE3
aei	TAE2
api	tinyProject1
api	tinyProject3
api	tinyProject2
ct	20220513T083900
ct	20220513T083900
ct	20220513T083900
et	20240513T083900
et	20240513T083900
et	20240513T083900
lt	20220513T083900
lt	20220513T083900
lt	20220513T083900
pi	5-20191210093452845
pi	5-20191210093452845
pi	5-20191210093452845
ri	TAE1
ri	TAE3
ri	TAE2
rn	Sensor1
rn	Sensor3
rn	Sensor2
rr	true
rr	true
rr	true
ty	2
ty	2
ty	2

변경할 저장 방식

Key	Value
TAE1	TAE1
TAE1	tinyProject1
TAE1	20220513T083900
TAE1	20240513T083900
TAE1	20220513T083900
TAE1	5-20191210093452845
TAE1	Sensor1
TAE1	true
TAE1	2
TAE2	TAE2
TAE2	tinyProject2
TAE2	20210513T083900
TAE2	20230513T083900
TAE2	20210513T083900
TAE2	5-20191210093452845
TAE2	Sensor2
TAE2	true
TAE2	2
TAE3	TAE3
TAE3	tinyProject3
TAE3	20200513T083900
TAE3	20220513T083900
TAE3	20200513T083900
TAE3	5-20191210093452845
TAE3	Sensor3
TAE3	true
TAE3	2

저장 구조 변경이 힘든 이유

TAE1 -> TAE3 -> TAE2 순서로 들어와도,

DB상에는 TAE1 -> TAE2 -> TAE3 순서로

저장

Node* Get_All_Sub()

Get_All_Sub

함수 기능 설명

- ✓ 모든 Sub를 Node 형태로 반환하는 함수.
- ✓ net는 int로 변환 후 반환
- ✓ Sub.db에 저장된 데이터가 없을 경우 NULL 반환

Input

없음

Return

ri : 23-2022040684653299304
rn : sub1
nu : http://223.131.176.101:3000/ct=json
net : 1
pi : 3-20220406084023203796

→

ri : 23-2021040684653299304
rn : sub2
nu : http://223.131.176.101:3000/ct=json
net : 8
pi : 3-20220406084023203796

Part 2

Node* Get_All_Sub() 동작방식

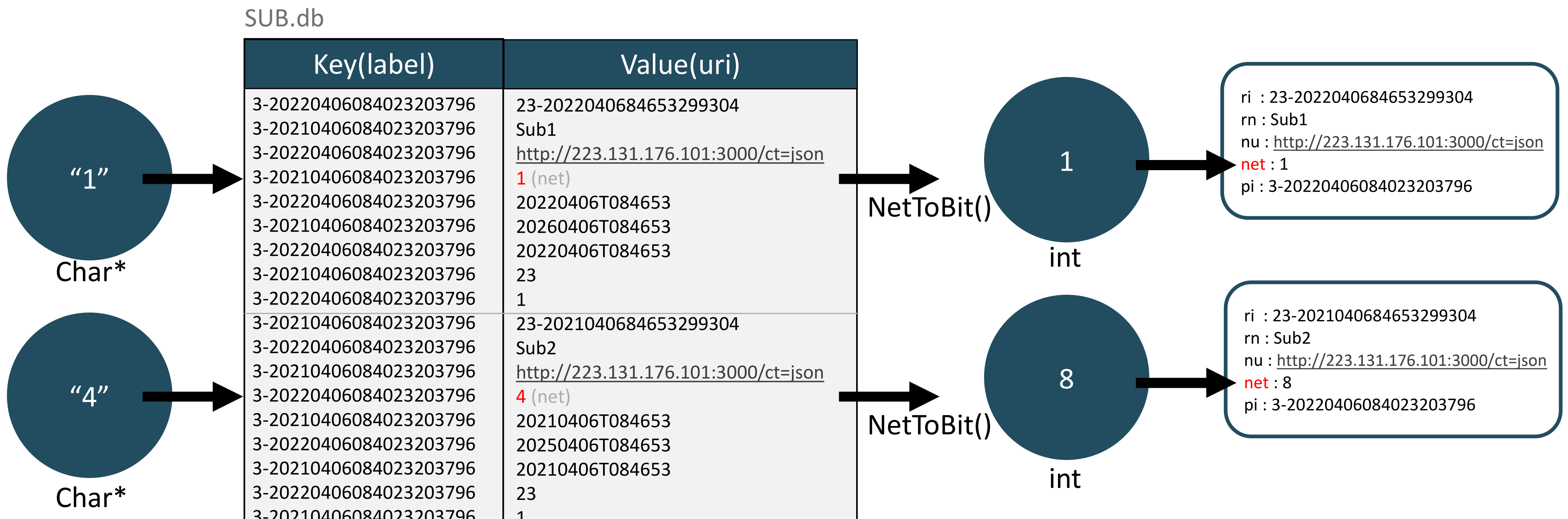
struct_size = 10

	Key(label)	Value(uri)	
sub1	3-20210406084023203796	23-202204068465329930	ri
	3-20210406084023203796	Sub1 http://223.131.176.101:3000/ct=json	rn nu
	3-20210406084023203796	1 (net)	sub_bit
	3-20210406084023203796	test/test/test/test1	net
	3-20210406084023203796	20220406T084653	ct
	3-20210406084023203796	20260406T084653	et
	3-20210406084023203796	20220406T084653	lt
	3-20210406084023203796	23	ty
	3-20210406084023203796	1	nct
sub2	3-20210406084023203796	23-202104068465329930	ri
	3-20210406084023203796	Sub2	rn
	3-20210406084023203796	http://223.131.176.101:3000/ct=json	nu
	3-20210406084023203796	8 (net)	sub_bit
	3-20210406084023203796	test/test/test/test2	net
	3-20210406084023203796	20210406T084653	ct
	3-20210406084023203796	20250406T084653	et
	3-20210406084023203796	20210406T084653	lt
	3-20210406084023203796	23	ty
	3-20210406084023203796	1	nct
sub3	3-20210406084023203796	23-202304068465329930	ri
	3-20210406084023203796	Sub3	rn
	3-20210406084023203796	http://223.131.176.101:3000/ct=json	nu
	3-20210406084023203796	256 (net)	sub_bit
	3-20210406084023203796	test/test/test/test3	net
	3-20210406084023203796	20230406T084653	ct
	3-20210406084023203796	20270406T084653	et
	3-20210406084023203796	20230406T084653	lt
	3-20210406084023203796	23	ty
	3-20210406084023203796	1	nct

```
...

while ((ret = dbcp->get(dbcp, &key, &data, DB_NEXT)) == 0) {
    if (strncmp(key.data, pi, key.size) == 0) {
        switch (idx) {
        case 0:
            node->ri = malloc(data.size);
            strcpy(node->ri, data.data);

            node->siblingRight = (SubNode*)malloc(sizeof(SubNode));
            node->siblingRight->siblingLeft = node;

            idx++;
            break;
        case 1:
            node->rn = malloc(data.size);
            strcpy(node->rn, data.data);

            idx++;
            break;
        case 2:
            node->nu = malloc(data.size);
            strcpy(node->nu, data.data);

            idx++;
            break;
        case 3:
            node->sub_bit = *(int*)data.data;

            idx++;
            break;
        case 4:
            node->pi = malloc(key.size);
            strcpy(node->pi, key.data);

            node = node->siblingRight;
            idx++;
            break;
        default:
            idx++;
            if (idx == struct_size) idx = 0;
        }
    }
}

...
```

Part 2

net 저장방식



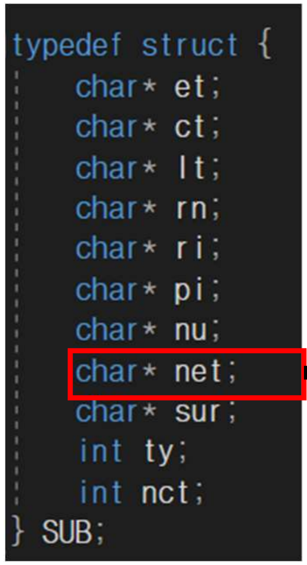
SUB.db

Key(label)	Value(uri)
3-20220406084023203796	23-20220406846532993 04
3-20210406084023203796	Sub1
3-20220406084023203796	http://223.131.176.101:3000/ct=json
3-20210406084023203796	1 (net)
3-20220406084023203796	20220406T084653
3-20210406084023203796	20260406T084653
3-20220406084023203796	20220406T084653
3-20210406084023203796	23
3-20220406084023203796	1
3-20210406084023203796	23-20210406846532993 04
3-20220406084023203796	Sub2
3-20210406084023203796	http://223.131.176.101:3000/ct=json
3-20220406084023203796	4 (net)
3-20210406084023203796	20210406T084653
3-20220406084023203796	20250406T084653
3-20210406084023203796	20210406T084653
3-20220406084023203796	23
3-20210406084023203796	1

ri : 23-20220406846532993 04
rn : Sub1
nu : http://223.131.176.101:3000/ct=json
net : 1
pi : 3-20220406084023203796

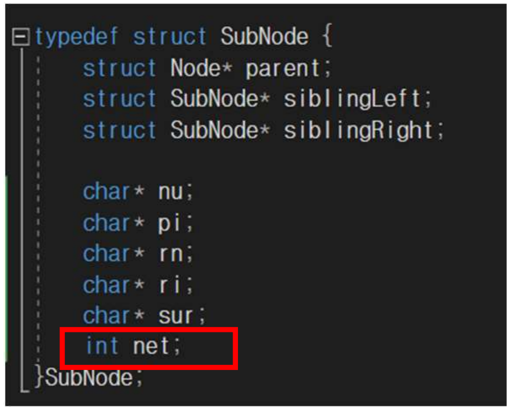
ri : 23-20210406846532993 04
rn : Sub2
nu : http://223.131.176.101:3000/ct=json
net : 8
pi : 3-20220406084023203796

net 저장방식


```c
typedef struct {
    char* et;
    char* ct;
    char* lt;
    char* rn;
    char* ri;
    char* pi;
    char* nu;
    char* net;
    char* sur;
    int ty;
    int nct;
} SUB;
```

SUB.db

Key(label)	Value(uri)
3-20220406084023203796	Sub1
3-20210406084023203796	23-2022040684653299304
3-20220406084023203796	http://223.131.176.101:3000/ct=json
3-20210406084023203796	1 (net)
3-20220406084023203796	20220406T084653
3-20210406084023203796	20260406T084653
3-20220406084023203796	20220406T084653
3-20210406084023203796	23
3-20220406084023203796	1
3-20210406084023203796	Sub2
3-20220406084023203796	23-2021040684653299304
3-20210406084023203796	http://223.131.176.101:3000/ct=json
3-20220406084023203796	4 (net)
3-20210406084023203796	20210406T084653
3-20220406084023203796	20250406T084653
3-20210406084023203796	20210406T084653
3-20220406084023203796	23
3-20210406084023203796	1

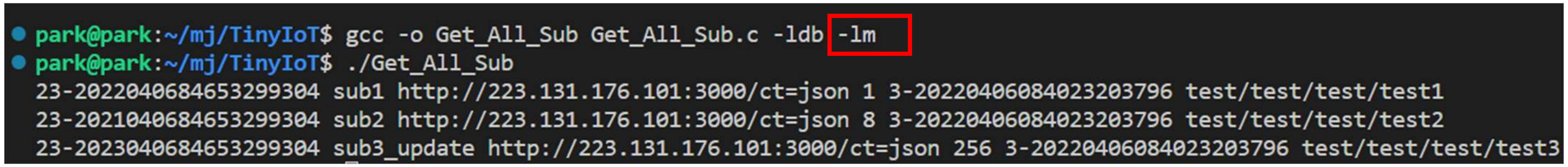

```c
typedef struct SubNode {
    struct Node* parent;
    struct SubNode* siblingLeft;
    struct SubNode* siblingRight;

    char* nu;
    char* pi;
    char* rn;
    char* ri;
    char* sur;
    int net;
}SubNode;
```

NetToBit()

```c
int NetToBit(char* net) {
    int netLen = strlen(net);
    int ret = 0;

    for (int i = 0; i < netLen; i++) {
        int exp = atoi(net + i);
        if (exp > 0) ret = (ret | (int)pow(2, exp - 1));
    }

    return ret;
}
```

undefined reference to `pow' 이슈

Ubuntu에서 C언어 math.h 헤더의 pow함수 사용 할 때
[gcc] undefined reference to `pow' 오류 발생



```
● park@park:~/mj/TinyIoT$ gcc -o Get_All_Sub Get_All_Sub.c -ldb -lm
● park@park:~/mj/TinyIoT$ ./Get_All_Sub
  23-2022040684653299304 sub1 http://223.131.176.101:3000/ct=json 1 3-20220406084023203796 test/test/test/test1
  23-2021040684653299304 sub2 http://223.131.176.101:3000/ct=json 8 3-20220406084023203796 test/test/test/test2
  23-2023040684653299304 sub3_update http://223.131.176.101:3000/ct=json 256 3-20220406084023203796 test/test/test/test3
```

컴파일 시 –lm 붙혀서 해결

https://forum.ubuntu-kr.org/viewtopic.php?t=16668