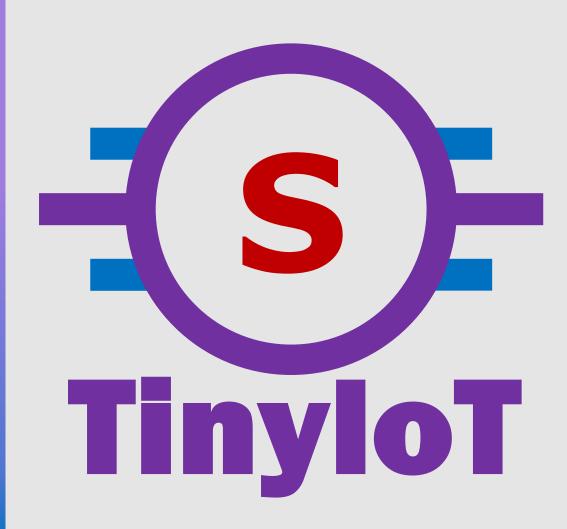
Berkeley DB for TinyloT

Sejong Univ.

Name: Park Minji

E-mail: iorw0224@gmail.com



Berkeley DB 이번주 진행 상황

- DB_Update_CNT
- 2 Subscription 관련 함수 구현
- 3 Notification 관련 함수 구현

4





DB_Update_CNT

Part 1 DB_Update_CNT

CNT* DB_Update_CNT(CNT* cnt_object)

cbs:-1 cbs : -1 cni:-1 cni:-1 ct: 202105T093154 ct: 202105T093154 et: 202105T093154 et: 202105T093154 lt: 202105T093154 lt: 202105T093154 pi: TAE2 pi: TAE2 ri: 3-20210513093154147745 ri: 3-20210513093154147745 rn: status2 update rn: status2 update st:-1 st:-1 ty:3 ty:3 Input Return 변경할 CNT 구조체 정보 변경할 CNT 구조체 정보

- ✓ cnt_object의 ri에 해당하는 오브젝트를 인자로 들어온 cnt_object의 값들로 변경 하는 함수
- ✓ 인자 이름을 AE_Update와는 다르게 cnt 대신 cnt object로 사용
- ✓ 인자로 들어온 ri가 존재하지 않으면 NULL 반화

Part 1 DB_Update_CNT(Update_AE와 동작 방식이 비슷함)

<UPDATE 전> AE.db

Cursor —

Find "TAE2"

세번째 index

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		

```
while ((ret = dbcp->get(dbcp, &key, &data, DB_NEXT)) == 0) {
    if (strncmp(key.data, "rn", key.size) == 0) {
        cnt_rn++;
        if (cnt_rn == idx) {
            data.size = strlen(ae->rn) + 1;
            dbcp->put(dbcp, &key, &data, DB_CURRENT);
        }
        if (strncmp(key.data, "pi", key.size) == 0) {
        cnt pi++;
```

AE2 수정

aei: TAE2_update api: tinyProject2_update ct: 20220817T053900 et: 20240817T053900 lt: 20220817T053900 pi: 5-20191210093452845

<mark>ri : TAE2</mark>

rn : Sensor2_update

rr : false ty : 8

<UPDATE 후> AE.db

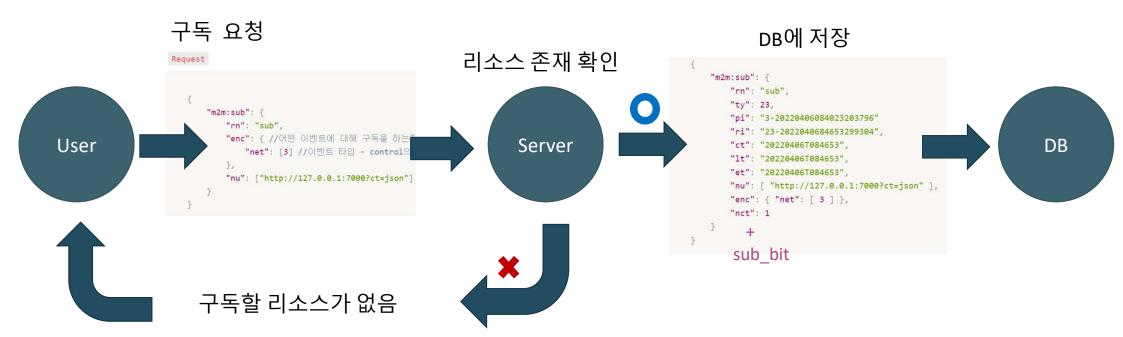
	VOI DI VI L	1 - 7 (2.00)
0) {	Key	Value
	aei	TAE1
<mark>현재 위치에</mark>	aei	TAE3
정	aei	TAE2_update
	api	tinyProject1
	api	tinyProject3
	api	tinyProject2_update
	ct	20220513T083900
	ct	20220513T083900
	ct	20220817T053900
	et	20240513T083900
	et	20240513T083900
	et	20240817T053900
	lt	20220513T083900
	lt	20220513T083900
	lt	20220817T053900
	pi	5-20191210093452845
	pi	5-20191210093452845
	pi	5-20191210093452845
	ri	TAE1
	ri	TAE3
Cursor—→	ri	TAE2
	rn	Sensor1
	rn	Sensor3
,	rn	Sensor2_update
	rr	true
	rr	true
	rr	false
	ty	2
	ty	2
	ty	8
		@Sachyool Vii Sachyool's PowerPr

©Saebyeol Yu. Saebyeol's PowerPoint

2

Subscription

Part 2 Subscription 절차



Part 2 Subscription 함수

int Subscription(SUB *sub_object)

rn:sub1

ri: 23-2022040684653299304

pi: 3-20220406084023203796

nu: http://223.131.176.101:3000/ct=json

net:3,1

ct: 20220406T084653 et: 20220406T084653 lt: 20220406T084653

ty:23 nct:1 sub bit:1

Input

구독할 SUB 구조체 정보

pi값에 NULL이 들어온 경우 -> 0 반환하고 저장되지 않음

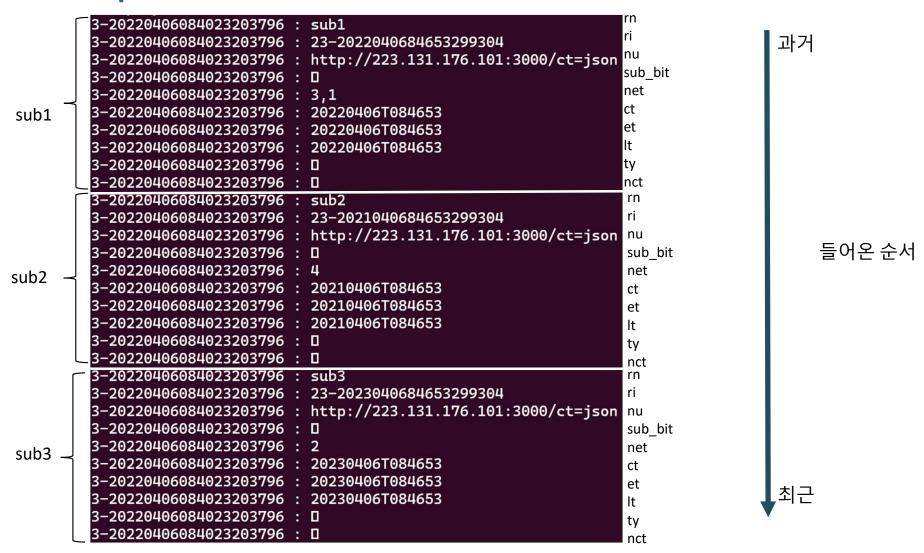
정상적으로 구독 정보 추가됨 -> 1 반환

Return

0 or 1

- ✓ SUB구조체를 인자로 받아 DB에 저장하는 함수
- ✓ DB 내부에서 pi를 키 값으로 설정
- ✓ pi값으로 NULL이 들어오면 에러 처리

Part 2 Subscription 함수저장형태

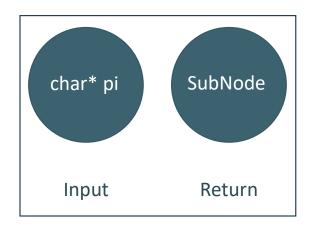


3

Notification

Part 3 Get Sub Pi함수

SubNode* Get_Sub_Pi(char* pi)



- ✓ pi에 해당하는 모든 sub들을 노드 형식으로 반환하는 함수
- ✓ 해당하는 sub가 DB에 없으면 NULL 반환

sub_bit(net 관련 값, 1-15)

rn ri nu pi

sub1 23-2022040684653299304 http://223.131.176.101:3000/ct=json 23-2022040684653299304 http://223.131.176.101:3000/ct=json 24-20220406084023203796 sub3 23-2023040684653299304 http://223.131.176.101:3000/ct=json 33-20220406084023203796

테스트 출력 형태

```
Part 3
           Get Sub Pi함수동작방식
                                                                  struct size = 10
         3-20220406084023203796
                                   sub1
                                                                        ri
         3-20220406084023203796
                                   23-2022040684653299304
                                                                         nu
                                   http://223.131.176.101:3000/ct=json
         3-20220406084023203796
                                                                         sub bit
         3-20220406084023203796
                                                                        net
         3-20220406084023203796
                                  3,1
sub1
                                   20220406T084653
                                                                         ct
         3-20220406084023203796
                                                                        et
         3-20220406084023203796
                                   20220406T084653
                                                                        lt
         3-20220406084023203796
                                   20220406T084653
         3-20220406084023203796
                                   ty
                                   П
         3-20220406084023203796
                                                                         nct
         3-20220406084023203796
                                   sub2
                                                                        rn
                                                                        ri
         3-20220406084023203796
                                   23-2021040684653299304
                                   http://223.131.176.101:3000/ct=json nu
         3-20220406084023203796
         3-20220406084023203796
                                                                        sub bit
                                  3-20220406084023203796
                                                                        net
sub2
         3-20220406084023203796
                                   20210406T084653
                                                                        ct
         3-20220406084023203796
                                   20210406T084653
                                                                        et
         3-20220406084023203796
                                   20210406T084653
                                                                        lt
         3-20220406084023203796
                                   ty
         3-20220406084023203796
                                                                        nct
                                   sub3
         3-20220406084023203796
         3-20220406084023203796
                                   23-2023040684653299304
                                                                        ri
         3-20220406084023203796
                                   http://223.131.176.101:3000/ct=json_nu
         3-20220406084023203796
                                  sub bit
         3-20220406084023203796
                                   2
                                                                        net
sub3
                                   20230406T084653
         3-20220406084023203796
                                                                        ct
         3-20220406084023203796
                                   20230406T084653
                                                                        et
         3-20220406084023203796
                                   20230406T084653
                                                                        lt
         3-20220406084023203796
                                   ty
         3-20220406084023203796
                                  П
                                                                        nct
```

```
while ((ret = dbcp->get(dbcp, &key, &data, DB NEXT)) == 0) {
     if (strncmp(kev.data, pi, kev.size) == 0) {
       switch (idx) {
       case 0:
          node->rn = malloc(data.size);
          strcpy(node->rn, data.data);
          node->siblingRight = (SubNode*)malloc(sizeof(SubNode));
          node->siblingRight->siblingLeft = node;
          idx++;
          break:
       case 1:
          node->ri = malloc(data.size);
          strcpv(node->ri, data.data);
          idx++;
          break:
       case 2:
          node->nu = malloc(data.size);
          strcpv(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 3 Get_Sub_Pi 함수 반환 값

SubNode* Get_Sub_Pi(char* pi)

반환되는 SubNode 정보

rn:sub1

ri: 23-2022040684653299304

nu: http://223.131.176.101:3000/ct=json

sub_bit:1

pi: 3-20220406084023203796

rn:sub2

ri: 23-2022040684653299304

nu: http://223.131.176.101:3000/ct=json

sub_bit:2

pi: 3-20220406084023203796

rn:sub3

ri: 23-2022040684653299304

nu: http://223.131.176.101:3000/ct=json

sub_bit:3

pi: 3-20220406084023203796

4

다음주수정사항

Part 4 다음주수정사항

- 1 Subscription 관련 함수 보완
- 2 Notification 관련 함수 보완
- **3** Filter Criteria



