

---

# Berkeley DB for TinyloT

---

Sejong Univ.

Name : Park Minji

E-mail : [iorw0224@gmail.com](mailto:iorw0224@gmail.com)



# Berkeley DB 이번주 진행 상황

---

- Node관련 수정 사항 적용
- pi에 해당하는 CIN들을 노드 형태로 반환하는 함수 Get\_CIN\_Pi(char\* pi) 구현
- Update\_AE(AE\* ae) 구현



[Github](#)



[Notion](#)

# Berkeley DB Node관련 수정사항 적용

구현 상 생기는 마지막 잉여 노드를 없애는 코드  
하지만 총 노드가 0개라면 node\_pi->siblingLeft 값이 존재하지 않기 때문에 node\_pi->siblingLeft->siblingRight 연산에서 Segmentation fault 발생!

```
void Free_Node(Node* node) {  
    free(node->ri);  
    free(node->rn);  
    free(node->pi);  
    free(node);  
}
```

```
if (node_pi->siblingLeft) node_pi->siblingLeft->siblingRight = NULL;  
else head = NULL;  
Free_Node(node_pi);  
node_ri = node_pi = node_rn = node_ty = NULL;
```

요건 새로운 Node 할당하는 부분인데 malloc으로만 동적 할당하면 Node 내에 변수들이 이상한 값들이 들어가서 트리 순회할 때 오류가 발생함!

```
Node* Create_Node(char* ri, char* rn, char* pi, ObjectType ty) {  
    Node* node = (Node*)malloc(sizeof(Node));  
    node->rn = (char*)malloc(sizeof(rn));  
    node->ri = (char*)malloc(sizeof(ri));  
    node->pi = (char*)malloc(sizeof(pi));  
    strcpy(node->rn, rn);  
    strcpy(node->ri, ri);  
    strcpy(node->pi, pi);  
    node->parent = NULL;  
    node->child = NULL;  
    node->siblingLeft = NULL;  
    node->siblingRight = NULL;  
    node->ty = ty;  
    if (strcmp(rn, "") && strcmp(rn, "TinyIoT")) {  
        fprintf(stderr, "WnCreate Tree NodeWn[rn] %sWn[ri] %sWn", node->rn, node->ri);  
    }  
    return node;  
}
```

동적 할당 후 초기화까지 진행해주는 별도 함수로 노드를 할당해주는 코드로 변경

```
node_pi->siblingRight = Create_Node("", "", "", 0);  
node_pi->siblingRight->siblingLeft = node_pi;  
node_pi = node_pi->siblingRight;
```

# Berkeley DB

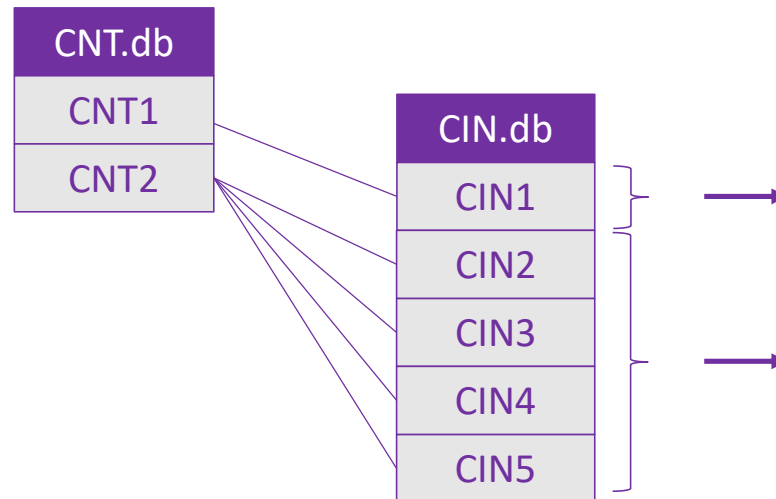
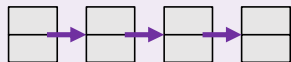
Get\_CIN\_Pi //pi에 해당하는 CIN들을 노드 형태로 반환하는 함수

함수 원형

```
Node* Get_CIN_Pi(char* pi)
```

✓ 인자 : parentID

✓ 반환형태 : Node



```
gcc -o Get_CIN_Pi Get_CIN_Pi.c -ldb  
./Get_CIN_Pi
```

Get\_CIN\_Period 출력 결과

부모 아이디 = CNT1

4-20220808T113154

부모 아이디 = CNT2

4-20220513T093154

4-20220808T093154

4-20220807T233154

4-20220807T113154

# Berkeley DB

Get\_CIN\_Pi //pi에 해당하는 CIN들을 노드 형태로 반환하는 함수

CIN.db

Key	Value
Pi	Cnt1
Pi	Cnt2
Pi	Cnt2
Pi	Cnt2
Pi	Cnt2

해당하는 pi

Key	Value
Pi	Cnt1
Pi	Cnt2
Pi	Cnt2
Pi	Cnt2
Pi	Cnt2

Cursor →

// 해당하는 오브젝트 배열에 1로 표시  
1 0 0 0 0 ← 첫 번째 오브젝트가 해당

```
// 해당하는 오브젝트 배열에 1로 표시
while ((ret = dbcp1->get(dbcp1, &key, &data, DB_NEXT)) == 0) {
    if (strncmp(key.data, "pi", key.size) == 0) {
        if (strcmp(pi, data.data) == 0)
            arr[idx] = 1;
        idx++;
    }
}
```

Pi = cnt1

Pi = cnt2

arr

1	0	0	0	0
---	---	---	---	---

idx

0	1	1	1	1
---	---	---	---	---

// 범위에 해당하는 오브젝트가 없을 때 -> NULL 반환

```
int sum = 0;
for (int i = 0; i < cnt; i++) {
    sum += arr[i];
}
if (sum == 0) {
    fprintf(stderr, "Data not exist\n");
    return NULL;
    exit(1);
}
```

arr

0	0	0	0	0
---	---	---	---	---

sum = 0

# Berkeley DB

AE\* Update\_AE(AE\* ae) //ae의 ri에 해당하는 오브젝트를 인자로 들어온 ae의 값들로 변경한다.

함수 원형

AE\* DB\_Update\_AE(AE\* ae)

: ae의 ri에 해당하는 오브젝트를 ae의 값들로 변경한다.

인자 : 수정할 AE

aei : TAE2\_update  
api : tinyProject2\_update  
ct : 20220817T053900  
et : 20240817T053900  
lt : 20220817T053900  
pi : 5-20191210093452845  
ri : TAE2  
rn : Sensor2\_update  
rr : false  
ty : 8

반환 형태 : 수정된 AE

aei : TAE2\_update  
api : tinyProject2\_update  
ct : 20220817T053900  
et : 20240817T053900  
lt : 20220817T053900  
pi : 5-20191210093452845  
ri : TAE2  
rn : Sensor2\_update  
rr : false  
ty : 8

```
// 수정할 오브젝트가 몇번째인지 찾기 위한 커서
DBC* dbcp0;
if ((ret = dbp->cursor(dbp, NULL, &dbcp0, 0)) != 0) {
    dbp->err(dbp, ret, "DB->cursor");
    exit(1);
}
// 인자로 들어온 ri가 몇 번째 index인지 찾기
while ((ret = dbcp0->get(dbcp0, &key, &data, DB_NEXT)) == 0) {
    if (strncmp(key.data, "ri", key.size) == 0) {
        idx++;
        if (strncmp(data.data, ae->ri, data.size) == 0) {
            cnt++; // update할 AE의 ri가 존재하면 cnt > 0
            break;
        }
    }
}
```



```
// 인자로 들어온 ri가 존재하지 않으면 NULL 반환
if (cnt == 0) {
    fprintf(stderr, "Data not exist\n");
    return NULL;
    exit(1);
}
```

# Berkeley DB AE\* Update\_AE(AE\* ae) DB 저장 형태

<UPDATE 전> AE.db

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

Cursor

Find "TAE2"  
세 번째 index

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

// 커서의 현재 위치에  
있는 값 수정

⋮

AE2 수정

aei : TAE2\_update  
api : tinyProject2\_update  
ct : 20220817T053900  
et : 20240817T053900  
lt : 20220817T053900  
pi : 5-20191210093452845  
ri : TAE2  
rn : Sensor2\_update  
rr : false  
ty : 8

<UPDATE 후> AE.db

Key	Value
aei	TAE1
aei	TAE3
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
ri	TAE2
rn	Sensor1
rn	Sensor3
rn	Sensor2_update
rr	true
rr	true
rr	false
ty	2
ty	2
ty	8

Cursor

# Berkeley DB AE\* Update\_AE(AE\* ae) 참고 자료

## 8.1.7.3 Subscription Update

Interoperability Test Description			
Identifier:	TD_M2M_NH_24		
Objective:	AE updates information about a subscription via subscription Update Request		
Configuration:	M2M_CFG_01		
References:	oneM2M TS-0001 [1], clause 10.2.10.4 oneM2M TS-0004 [2], clause 7.4.8.2.3		
Pre-test conditions:	<ul style="list-style-type: none"><li>AE has created an Application Entity resource &lt;AE&gt; on Registrar CSE</li><li>AE has created a container resource &lt;container&gt; on Registrar CSE</li><li>AE has created a subscription resource &lt;subscription&gt; on Registrar CSE</li></ul>		
Test Sequence			
Step	RP	Type	Description
1		Stimulus	AE is requested to send a subscription Update Request to update the lifetime of the resource.
2	Mca	PRO Check Primitive	<ul style="list-style-type: none"><li>op = 3 (Update)</li><li>to = {CSEBaseName}/URI of &lt;Subscription&gt; resource</li><li>fr = AE-ID</li><li>rqi = (token-string)</li><li>pc = Serialized representation of updated &lt;Subscription&gt; resource</li></ul>
3		IOP Check	Check if possible that the <subscription> resource is updated in Registrar CSE
4	Mca	PRO Check Primitive	<ul style="list-style-type: none"><li>rsc = 2004 (Updated)</li><li>rqi = (token-string) same as received in request message</li><li>pc = Serialized representation of &lt;Subscription&gt; resource</li></ul>
5		IOP Check	AE indicates successful operation
IOP Verdict			
PRO Verdict			