

F7-010 【程式設計(一)】HW10

Function 4 : segment()

```
void segment(int count){
    for (int i = 0; i < count; i++) {
        char *slash_pos = strchr(routing_table[i], '/');
        if (slash_pos) {
            int prefix = atoi(slash_pos + 1);
            prefix_insert(routing_table[i], prefix);
        }
    }
}
```

說明：

- void segment 傳入 int count (在 main function 傳入資料為在 Function 1 輸出的 PrefixesCount，為檔案中的 ip 數量。)
- 進行迴圈，從 0 執行到 count 次，也就是覆蓋所有的 ip。
- 透過 strchr 函數找到 routing_table[i] 第一次出現「/」的位置，存儲該記憶體位置到「slash_pos」指標。
- 如果有找到「slash_pos」，則 prefix 是「slash_pos」以後的數值。調用 prefix_insert 函數，並傳入 routing_table[i] (該 ip) 以及 prefix (該數值)。(prefix_insert 函數將在之後的 function 實作)

Function 5 : prefix_insert()

```
void prefix_insert(char *ip, int prefix){
    struct hash *new_node = (struct hash *)malloc(sizeof(struct hash));
    strcpy(new_node->ip, ip);
    new_node->next = hash[prefix].next;
    hash[prefix].next = new_node;
}
```

說明：

- void prefix_insert 傳入 ip 地址以及 prefix 數值。
- 創建一個 hash 鏈結（先前已定義結構於程式）為 new_node，並將其記憶體擴充 sizeof(struct hash) 單位。
- 將 new_node 的 ip 設為傳入的 ip。
- 將 new_node 的下一位指向 hash[prefix]的下一位。
- 而 hash[prefix]的下一位為 new_node 本身。
- 平均插入秒數：<0.000001 秒

Function 6 : prefix_delete()

```
void prefix_delete(char file[]) {  
    FILE *F = fopen(file, "r");  
    char line[50];  
    while (fgets(line, 50, F)) {  
        char *slash_pos = strchr(line, '/');  
        if (slash_pos) {  
            int prefix = atoi(slash_pos + 1);  
            struct hash *prev = &hash[prefix];  
            struct hash *cur = hash[prefix].next;  
            while (cur) {  
                if (strcmp(cur->ip, line) == 0) {  
                    prev->next = cur->next;  
                    free(cur);  
                    break;  
                }  
                prev = cur;  
                cur = cur->next;  
            }  
        }  
    }  
    fclose(F);  
}
```

說明：

- void prefix_delete 傳入要刪除的清單檔案。
- 開啟該檔案。
- 進行讀取檔案，逐行讀取，存入 char line[50]中。
- 搜尋 prefix 的數值並存入，再到該 prefix 的 hash 鏈結中尋找符合欲刪除的 ip（使用 strcmp 比較），若找到的話將前一個 next 直接指向下一個，即完成刪除該指定 ip 於 hash 鏈結之動作。
- 關閉檔案。
- 平均刪除秒數：0.000203 秒

Function 7 : search()

```
void search(char file[]) {
    FILE *F = fopen(file, "r");
    char line[50];
    while (fgets(line, 50, F)) {
        char *slash_pos = strchr(line, '/');
        if (slash_pos) {
            int prefix = atoi(slash_pos + 1);
            struct hash *cur = hash[prefix].next;
            int found = 0;
            while (cur) {
                if (strcmp(cur->ip, line) == 0) {
                    printf("IP %s Found.\n", line);
                    found = 1;
                    break;
                }
                cur = cur->next;
            }
            if (!found) {
                printf("IP %s Not Found.\n", line);
            }
        }
    }
    fclose(F);
}
```

說明：

- void search 為尋找指定 ip 是否在儲存的資料庫中。傳入檔案表，逐一檢查。
- 開啟檔案。
- 同於 function 6 的動作，逐行讀取檔案要尋找的 ip，先判斷他的 prefix，然後再進入該 prefix 的 hash 表中檢索。
- 使用 strcmp 去尋找 hash Linkedlist 是否有完全相同於欲尋找 ip 的 ip 地址，如果為 0（表示完全沒差異），就輸出「IP {ip} 已被找到」，反之，輸出「IP {ip} 沒被找到」。
- 關閉檔案。
- 平均搜尋秒數：0.000200 秒