

C Programming

Day20.B

2017.11.28

Introduction to Data Structures

Copyright (c) 2015 - 2017 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

```
#include <stdio.h>

struct aaa {
    int data; // 4-byte
    struct aaa *next; // 8-byte
} ;

typedef struct node node;

struct node {
    int data; // 4-byte
    node *next; // 8-byte
} ;

int main(void) {
    struct aaa var1;
    node var2;

    var1.data = 111;
    var1.next = NULL;

    printf("var1.next = %p \n", var1.next);
    printf("var1.data = %d \n", var1.data);

    printf("sizeof(var1) = %ld \n", sizeof(var1));

    var2.data = 111;
    var2.next = NULL;

    printf("var2.next = %p \n", var2.next);
    printf("var2.data = %d \n", var2.data);

    printf("sizeof(var2) = %ld \n", sizeof(var2));
}
```

```
typedef struct node node;

struct node {
    int data; // 4-byte
    node *next; // 8-byte
} ;

int main(void) {
    node n1, n2, n3;
    node *sp = NULL;
    node *ep = NULL;
    node *p = NULL;

    sp = &n1;
    ep = &n3;

    n1.data = 111;
    n1.next = &n2;

    n2.data = 222;
    n2.next = &n3;

    n3.data = 333;
    n3.next = NULL;

    printf("&n1 = %p sizeof(n1)=%ld \n", &n1, sizeof(n1));
    printf("&n2 = %p sizeof(n2)=%ld \n", &n2, sizeof(n2));
    printf("&n3 = %p sizeof(n3)=%ld \n", &n3, sizeof(n3));

    printf("sp = %p \n", sp);
    printf("ep = %p \n", ep);
}
```

```
printf("n1.data = %d \n", n1.data);
printf("n1.next = %p \n", n1.next);

printf("n2.data = %d \n", n2.data);
printf("n2.next = %p \n", n2.next);

printf("n3.data = %d \n", n3.data);
printf("n3.next = %p \n", n3.next);

printf("-----\n");

p = sp;
printf("p = sp \n");
printf("\t\t p->data = %d \n", p->data);
printf("\t\t p->next = %p \n", p->next);

p = p->next;
printf("p = p->next \n");
printf("\t\t p->data = %d \n", p->data);
printf("\t\t p->next = %p \n", p->next);

p = p->next;
printf("p = p->next \n");
printf("\t\t p->data = %d \n", p->data);
printf("\t\t p->next = %p \n", p->next);

p = p->next;
printf("p = p->next \n");

printf("-----\n");

p = sp;

while (p) {
    printf("\t\t p->data = %d \n", p->data);
    printf("\t\t p->next = %p \n", p->next);

    p = p->next;
}
}
```

```
&n1 = 0x7ffd7c981fc0  sizeof(n1)=16
&n2 = 0x7ffd7c981fd0  sizeof(n2)=16
&n3 = 0x7ffd7c981fe0  sizeof(n3)=16
sp  = 0x7ffd7c981fc0
ep  = 0x7ffd7c981fe0
n1.data = 111
n1.next = 0x7ffd7c981fd0
n2.data = 222
n2.next = 0x7ffd7c981fe0
n3.data = 333
n3.next = (nil)
-----
p = sp
                p->data = 111
                p->next = 0x7ffd7c981fd0
p = p->next
                p->data = 222
                p->next = 0x7ffd7c981fe0
p = p->next
                p->data = 333
                p->next = (nil)
p = p->next
-----
                p->data = 111
                p->next = 0x7ffd7c981fd0
                p->data = 222
                p->next = 0x7ffd7c981fe0
                p->data = 333
                p->next = (nil)
```