



# Programming with C I

Fangtian Zhong  
CSCI 112

Gianforte School of Computing  
Norm Asbjornson College of Engineering  
E-mail: [fangtian.zhong@montana.edu](mailto:fangtian.zhong@montana.edu)

# Linked Lists

## ➤ **linked list**

- a sequence of nodes in which each node but the last contains the address of the next node

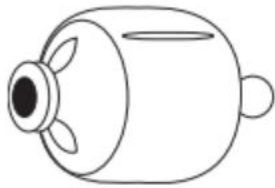
## ➤ **empty list**

- a list of no nodes
- represented in C by the pointer NULL, whose value is zero

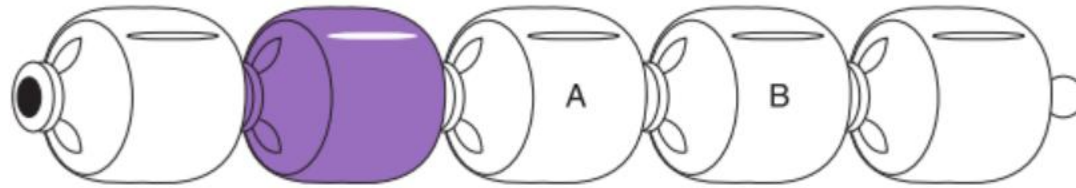
## ➤ **list head**

- the first element in a linked list

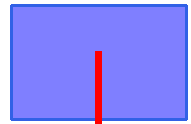
# Figure Children's Pop Beads in a Chain



Pop bead



Chain of Pop beads



current volts linkp

A C \0	115	→
--------	-----	---

current volts linkp

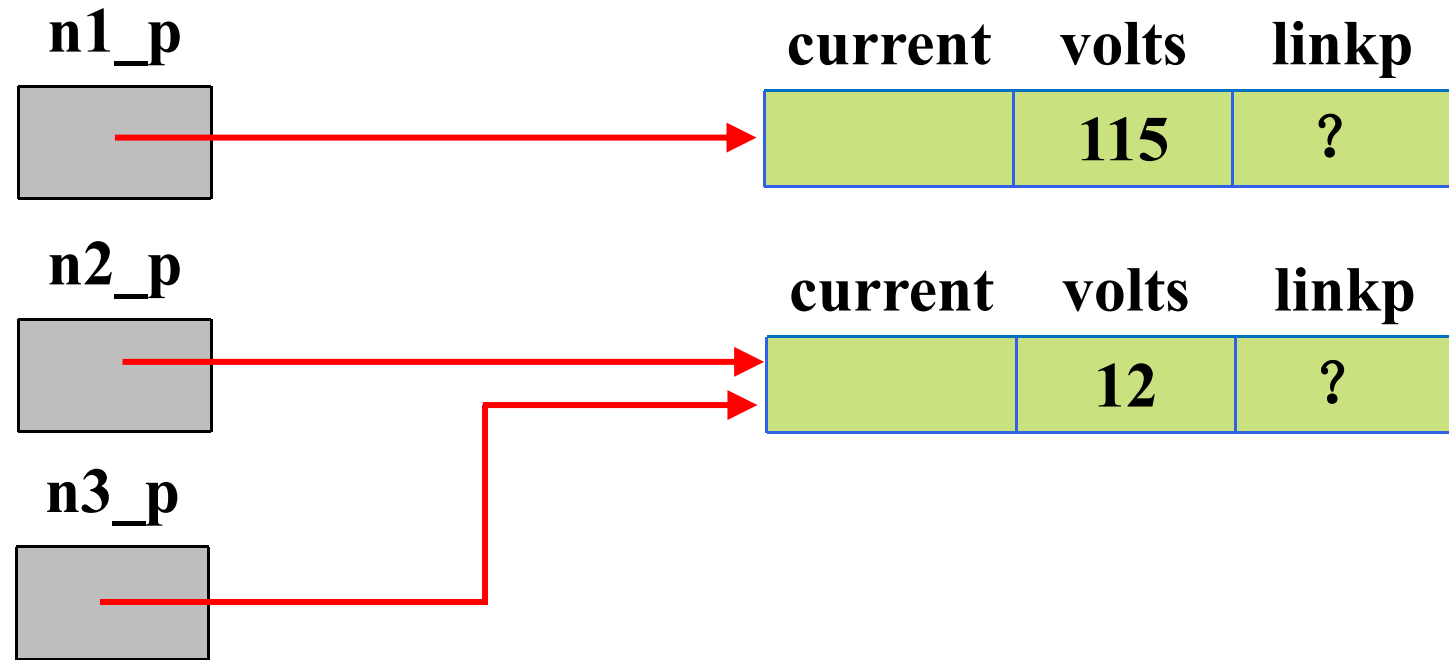
D C \0	12	→
--------	----	---

current volts linkp

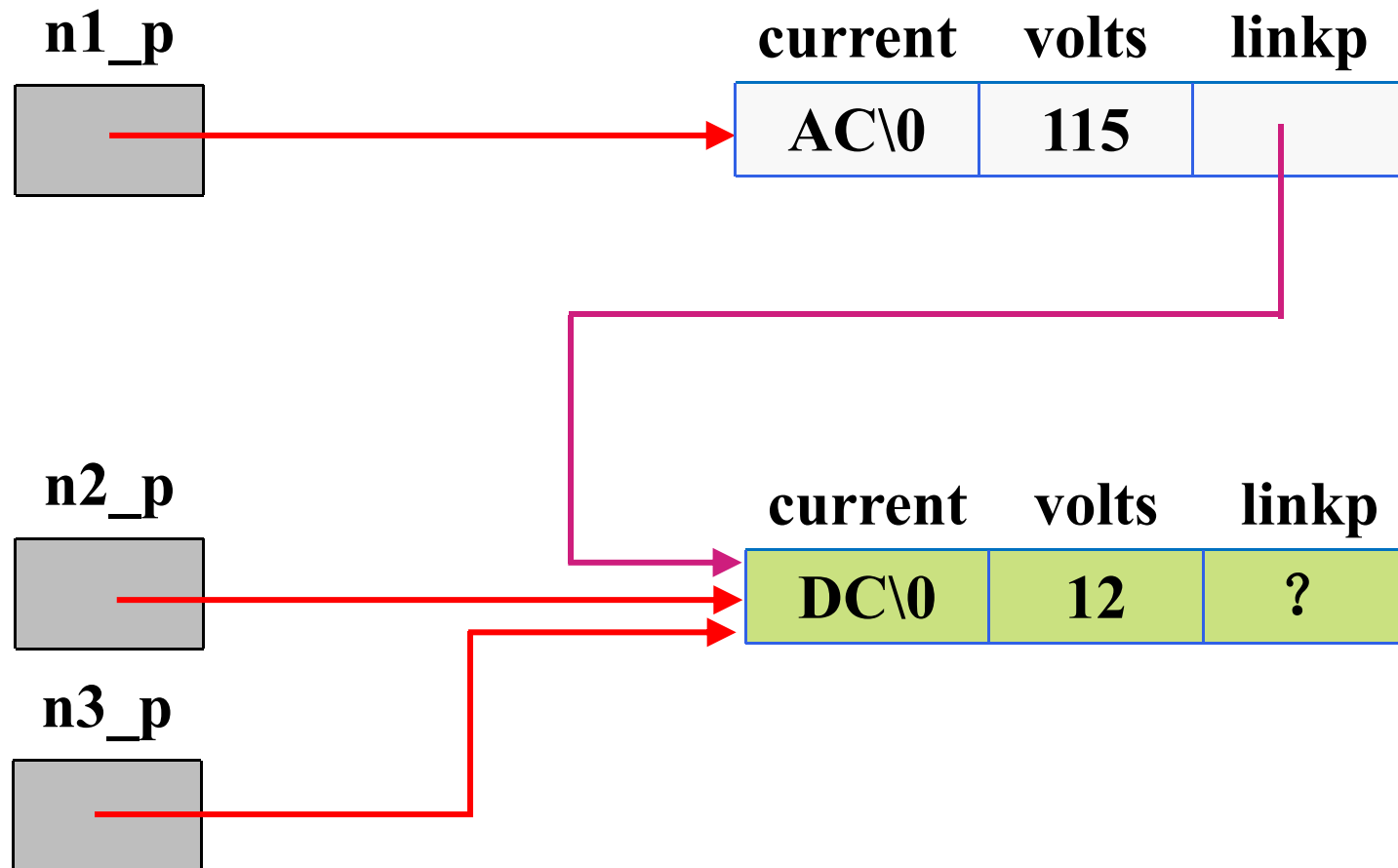
A C \0	220	/
--------	-----	---

## Figure Multiple Pointers to the Same Structure

```
node_t *n1_p, *n2_p, *n3_p;  
n1_p = (node_t *)malloc(sizeof (node_t));  
strcpy(n1_p->current, "AC");  
n1_p->volts = 115;  
n2_p = (node_t *)malloc(sizeof (node_t));  
strcpy(n2_p->current, "DC");  
n2_p->volts = 12;  
n3_p = n2_p;
```



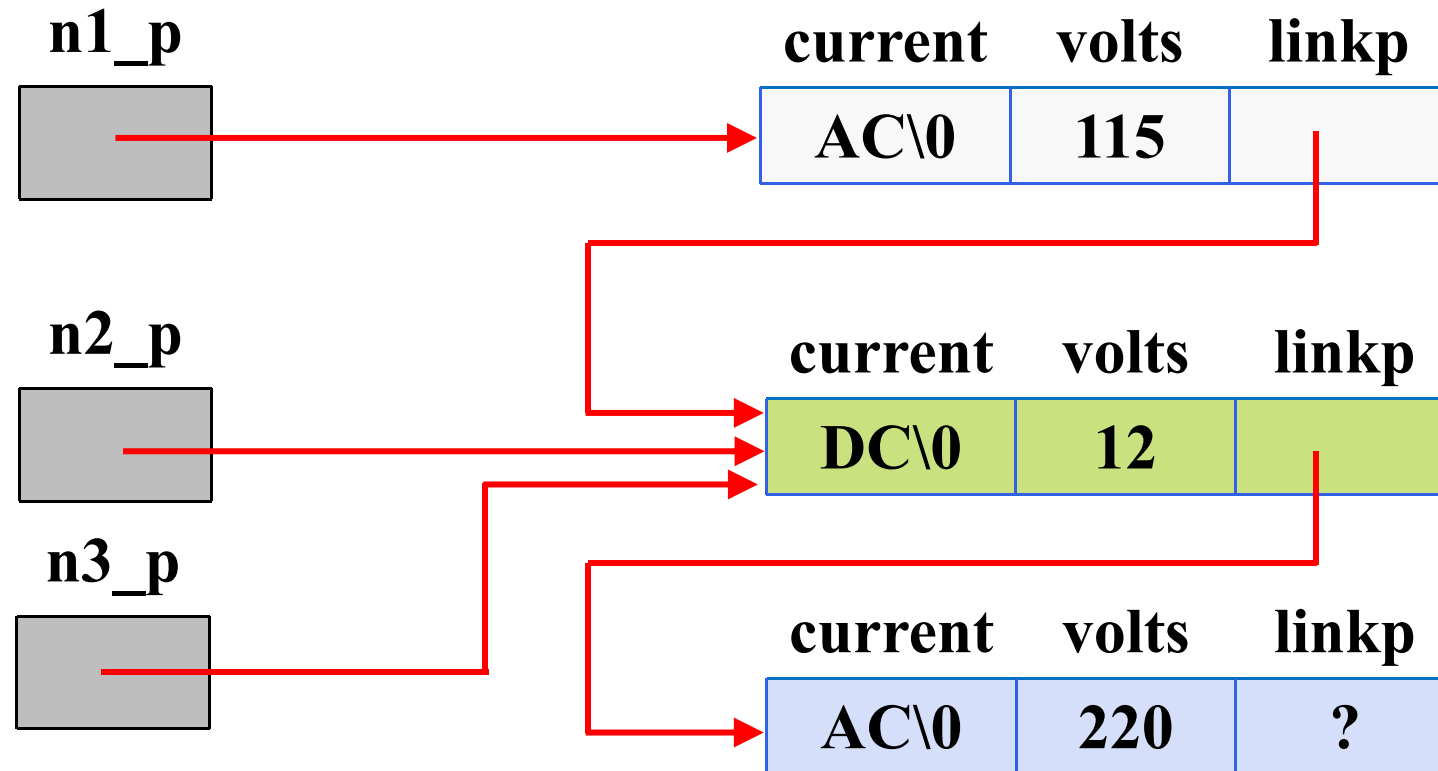
## Figure Linking Two Nodes



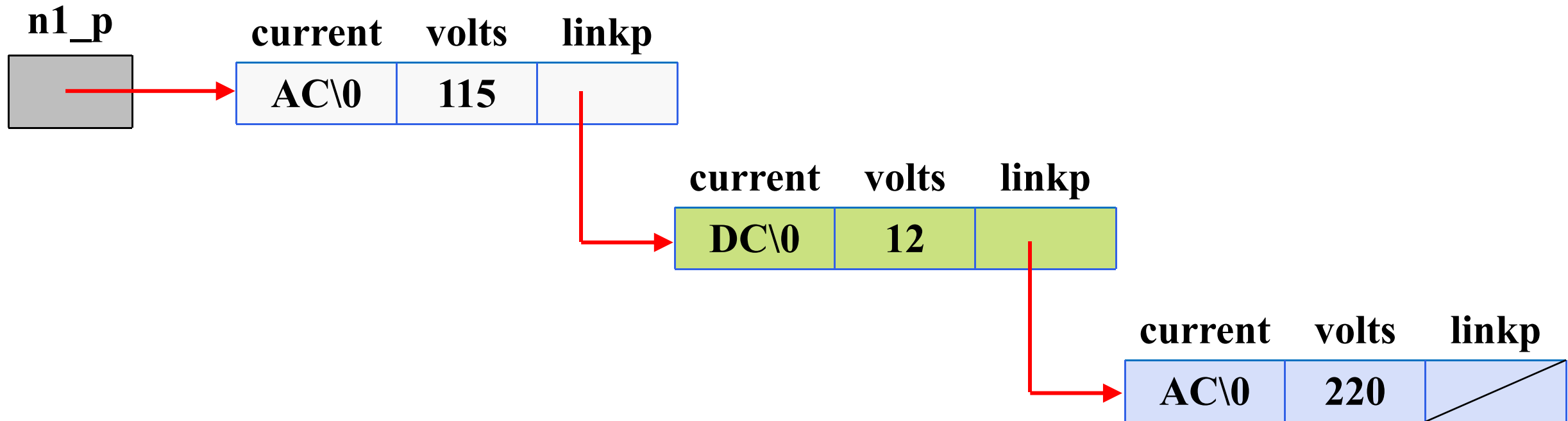
## Table Analyzing the Reference n1\_p->linkp->volts

Section of Reference	Meaning
n1_p->linkp	Follow the pointer in n1_p to a structure and select the <b>linkp</b> component
linkp->volts	Follow the pointer in the <b>linkp</b> component to another structure and select the <b>volts</b> component.

## Figure Three-Node Linked List with Undefined Final Pointer



## Figure Three-Element Linked List Accessed Through n1\_p





```

digit* create_new_digit(int d) {
    digit* new = malloc(sizeof(digit));
    new->d = d;
    new->next = NULL;
    return(new);
}

int main(void) {
    digit* head;
    head = create_new_digit(1);
    head->next = create_new_digit(2);
    head->next->next = create_new_digit(3);
}

```

## Stack memory

main  
 digit\* head: ??

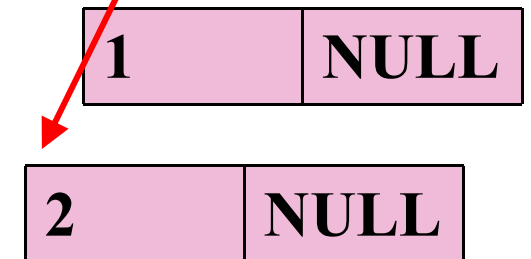
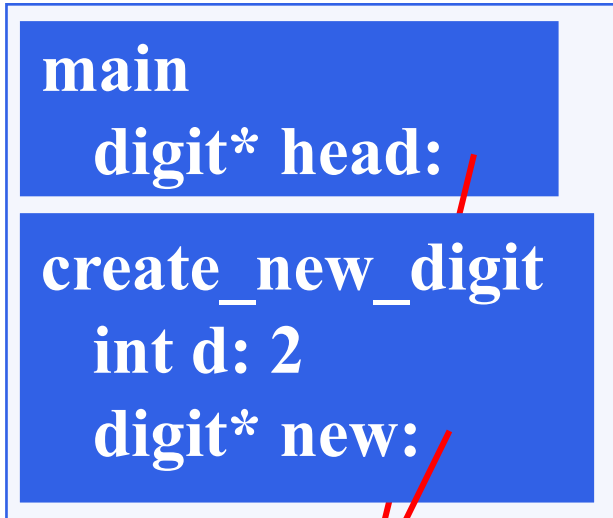
create\_new\_digit  
 int d: 1  
 digit\* new:



## Heap memory

```
digit* create_new_digit(int d) {  
    digit* new = malloc(sizeof(digit));  
    new->d = d;  
    new->next = NULL;  
    return(new);  
}  
  
int main(void) {  
    digit* head;  
    head = create_new_digit(1);  
    head->next = create_new_digit(2);  
    head->next->next = create_new_digit(3);  
}
```

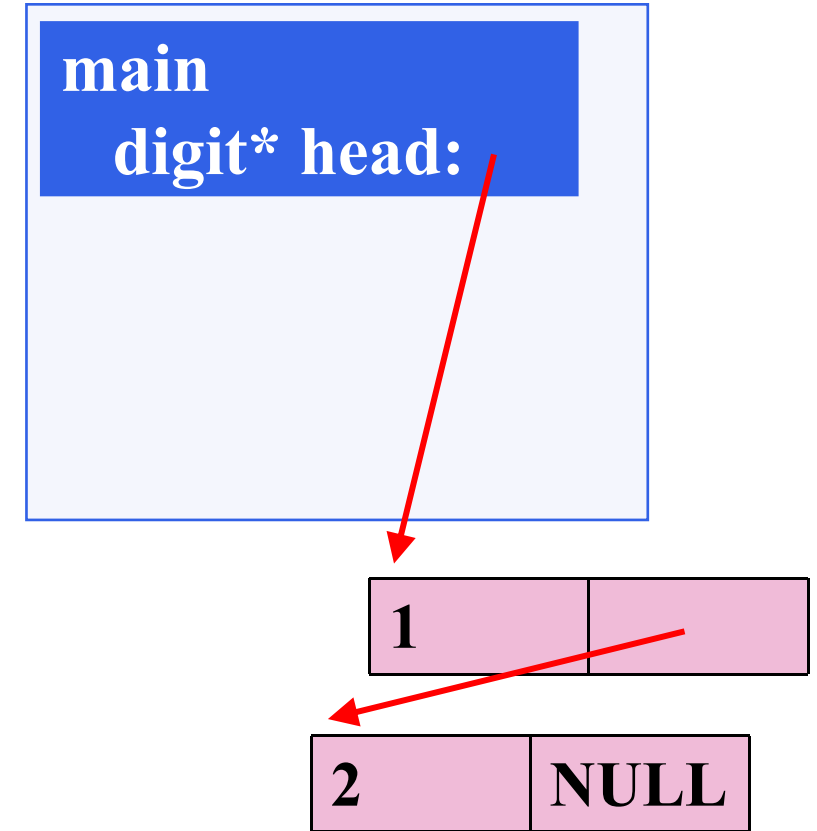
## Stack memory



## Heap memory

```
digit* create_new_digit(int d) {  
    digit* new = malloc(sizeof(digit));  
    new->d = d;  
    new->next = NULL;  
    return(new);  
}  
  
int main(void) {  
    digit* head;  
    head = create_new_digit(1);  
    head->next = create_new_digit(2);  
    head->next->next = create_new_digit(3);  
}
```

## Stack memory

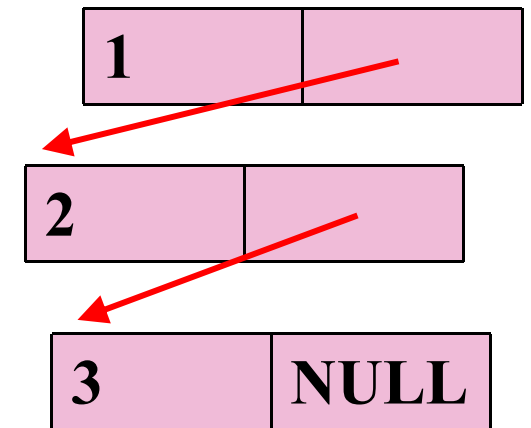
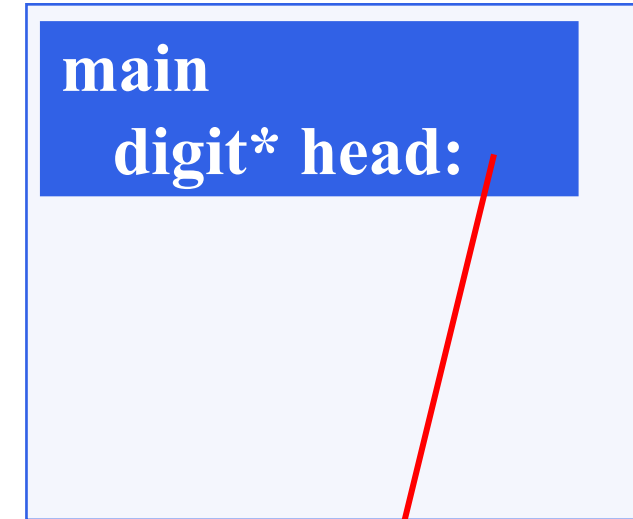


## Heap memory

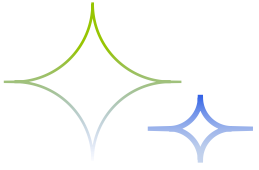
```
digit* create_new_digit(int d) {  
    digit* new = malloc(sizeof(digit));  
    new->d = d;  
    new->next = NULL;  
    return(new);  
}
```

```
int main(void) {  
    digit* head;  
    head = create_new_digit(1);  
    head->next = create_new_digit(2);  
    head->next->next = create_new_digit(3);  
}
```

## Stack memory



## Heap memory



# THE END

Fangtian Zhong  
CSCI 112