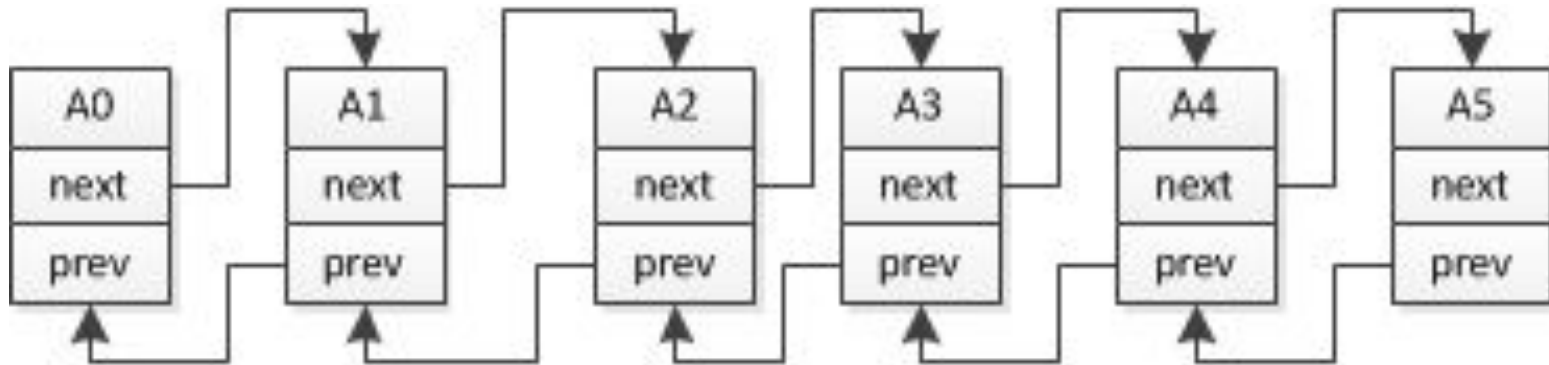
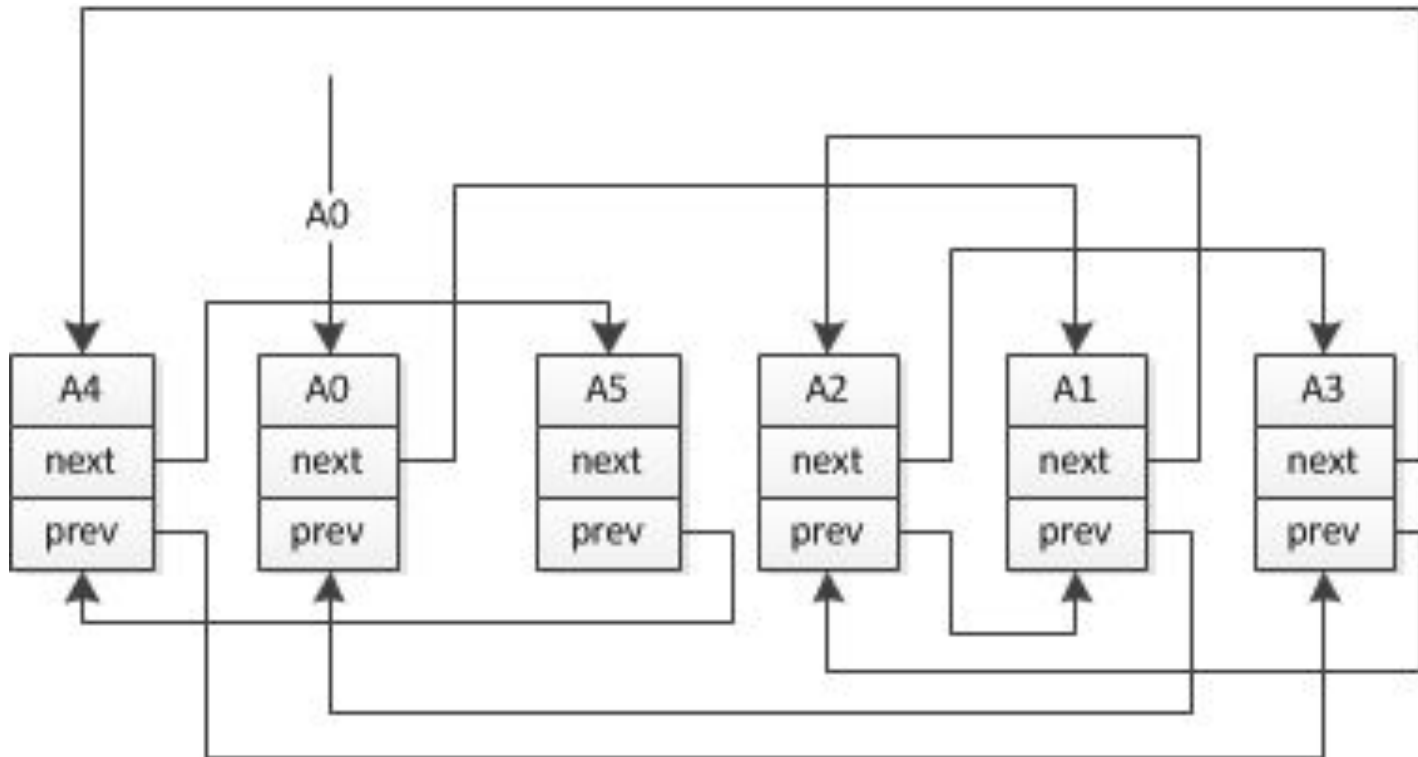


Двусвязные списки

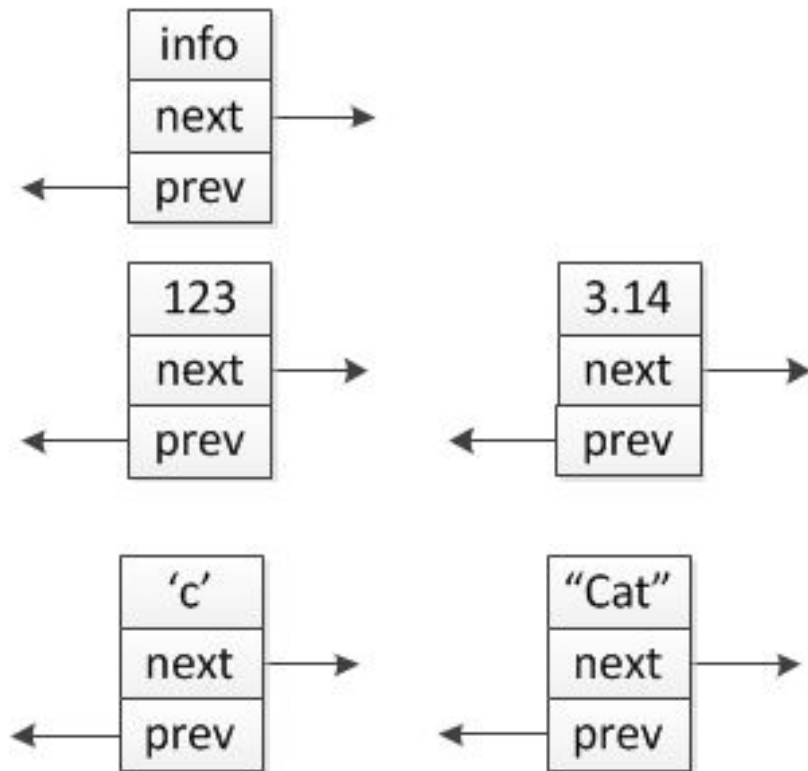
Идея списка



Идея списка



Узел списка и его содержимое



Узел списка (код)

```
struct node{  
    int info;  
    node* next;  
    node* prev;  
};
```

Список руками

```
struct node{
    int info;
    node *next;
    node *prev;
};

int main() {
    node *first, * second;
    first = new node;
    first->info = 10;
    second = new node;
    second->info = 20;
```

```
    first->next = second;
    second->next = NULL;
    first->prev = NULL;
    second->prev = first;

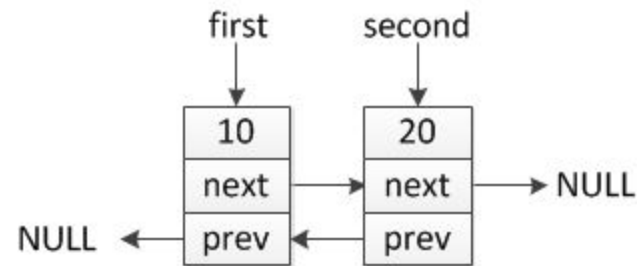
    cout << first->info << endl;
    cout << second->info << endl;
    cout << first->next->info << endl;
    cout << second->next->info << endl;
    cout << first->prev->info << endl;
    cout << second->prev->info << endl;
}
```

Список руками

```
struct node{
    int info;
    node *next;
    node *prev;
};
int main() {
    node *first, * second;
    first = new node;
    first->info = 10;
    second = new node;
    second->info = 20;
```

```
first->next = second;
second->next = NULL;
first->prev = NULL;
second->prev = first;

cout << first->info << endl;
cout << second->info << endl;
cout << first->next->info << endl;
cout << second->next->info << endl;
cout << first->prev->info << endl;
cout << second->prev->info << endl;
}
```



Список руками 2

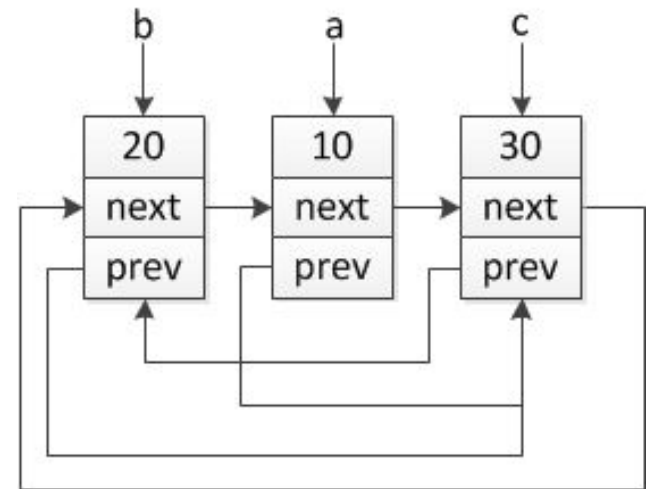
```
node *a, *b, *c;  
a = new node;  
a->info = 10;  
b = new node;  
b->info = 20;  
c = new node;  
c->info = 30;  
a->next = c; a->prev=c;  
b->next = a; b->prev=c;  
c->next = b; c->prev=b;
```

```
cout << a->next->info << endl;  
cout << a->next->next->info << endl;  
cout << b->next->info << endl;  
cout << b->next->next->info << endl;  
cout << c->next->info << endl;  
cout << c->next->next->info << endl;  
  
cout << a->prev->info << endl;  
cout << a->prev->prev->info << endl;  
cout << b->prev->info << endl;  
cout << b->prev->prev->info << endl;  
cout << c->prev->info << endl;  
cout << c->prev->prev->info << endl;
```


Список руками 2

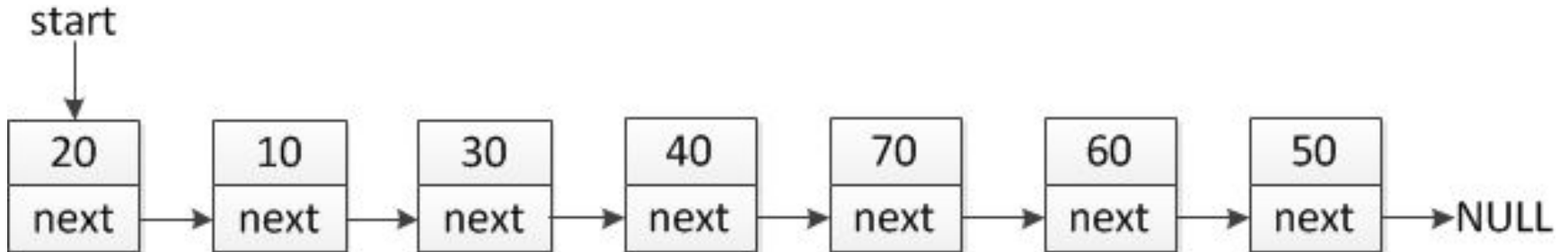
```
node *a, *b, *c;  
a = new node;  
a->info = 10;  
b = new node;  
b->info = 20;  
c = new node;  
c->info = 30;  
a->next = c;  
b->next = a;  
c->next = b;
```

```
cout << a->next->info << endl;  
cout << a->next->next->info << endl;  
cout << b->next->info << endl;  
cout << b->next->next->info << endl;  
cout << c->next->info << endl;  
cout << c->next->next->info << endl;  
cout << a->prev->info << endl;  
cout << a->prev->prev->info << endl;  
cout << b->prev->info << endl;  
cout << b->prev->prev->info << endl;  
cout << c->prev->info << endl;  
cout << c->prev->prev->info << endl;
```



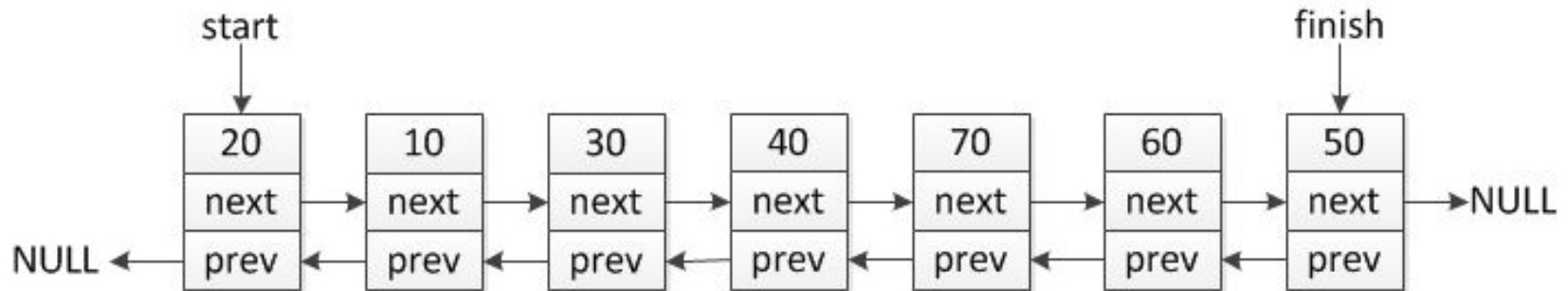
Вывод списка

```
for(node* cur = start; cur!=NULL; cur = cur->next){  
    cout << cur->info << endl;  
}
```

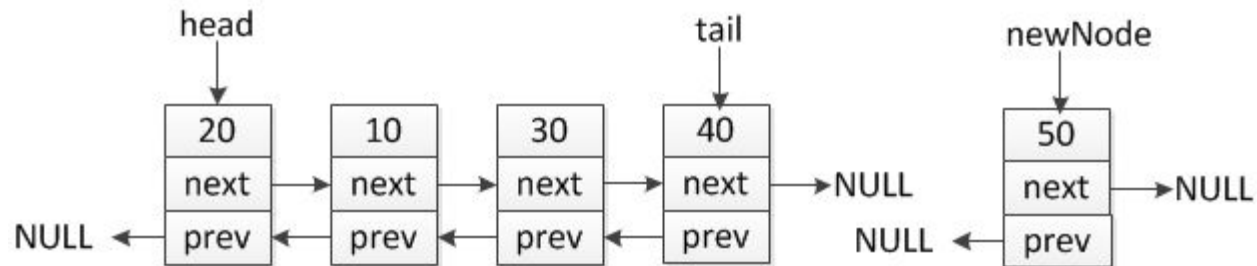


Вывод списка

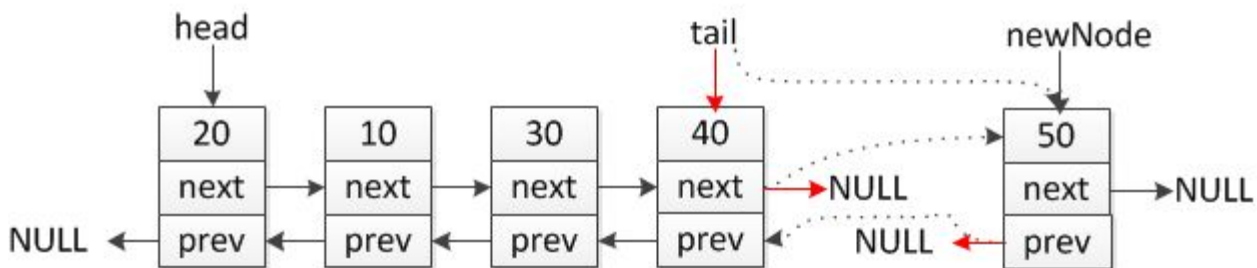
```
for(node* cur = start; cur!=NULL; cur = cur->next){  
    cout << cur->info << endl;  
}  
for(node* cur = finish; cur!=NULL; cur = cur->prev){  
    cout << cur->info << endl;  
}
```



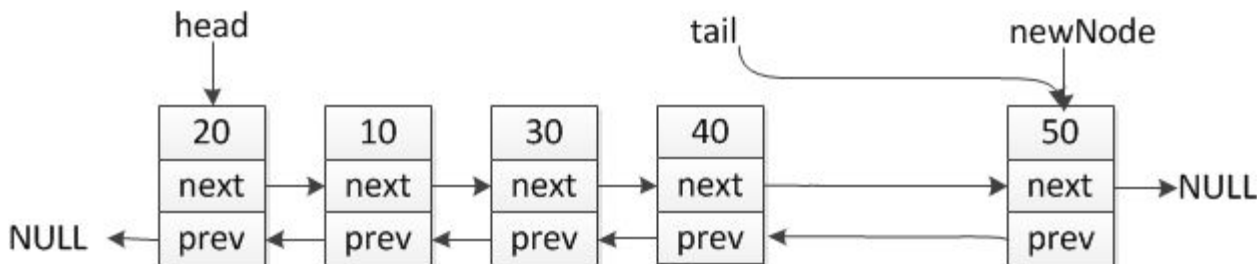
Дополнение списка



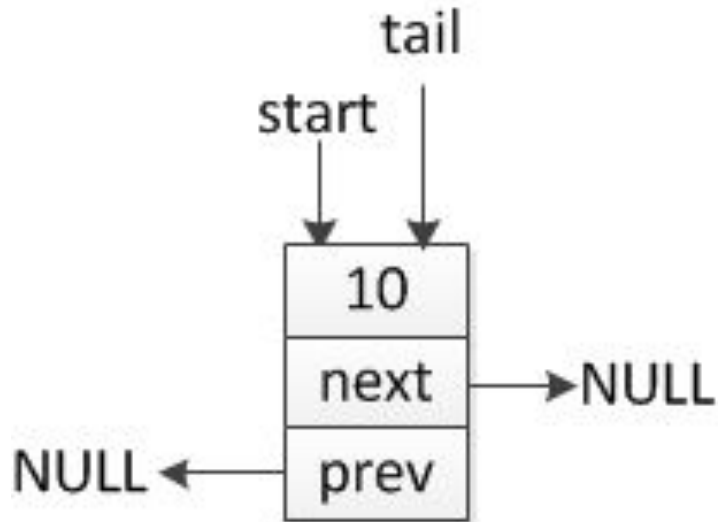
```
newNode = new node;  
newNode->info = 50;  
newNode->next = NULL;  
newNode->prev = NULL;
```



```
tail->next = newNode;  
newNode->prev = tail;  
tail = newNode;
```



Создание списка с нуля



```
node *newNode = new node;  
newNode->info = 10;  
newNode->next = NULL;  
newNode->prev = NULL;  
start = newNode;  
tail = newNode;
```

Ввод-обработка-вывод

```
node *head=NULL, *tail=NULL, *newNode; int x;
while(cin>>x){
    newNode = new node;
    newNode->info = x;
    newNode->next = NULL;
    newNode->prev = NULL;
    if(tail==NULL){
        head = newNode;
        tail = newNode;
    } else {
        tail->next = newNode;
        newNode->prev = tail;
        tail = newNode;
    }
}
```

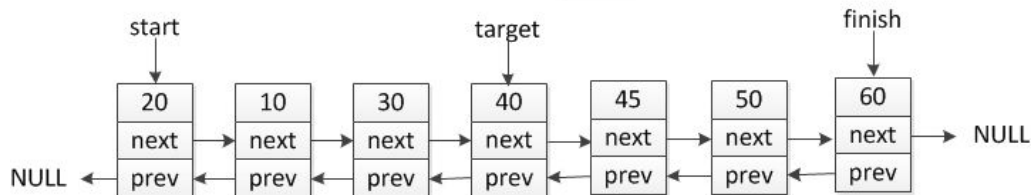
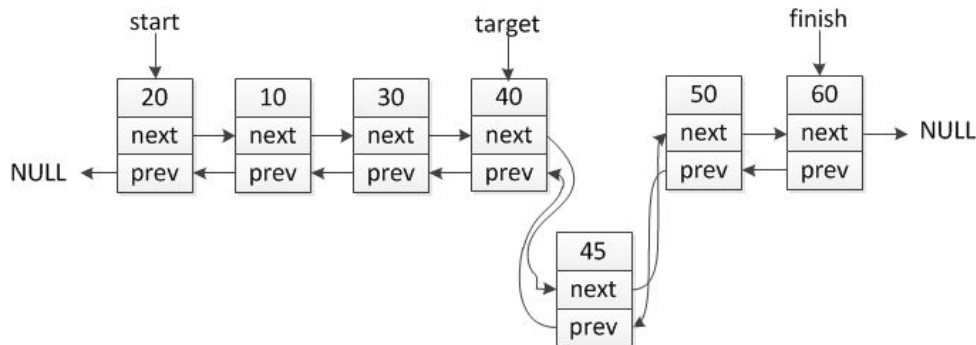
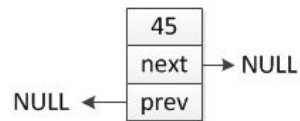
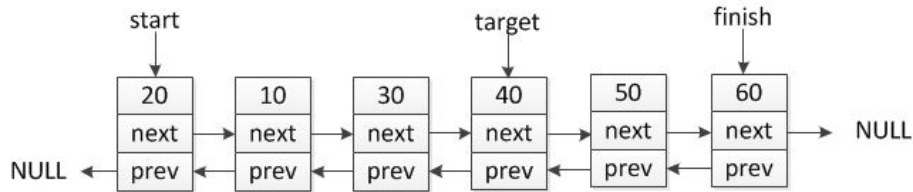
Ввод-обработка-вывод

```
for(node* cur = head; cur!=NULL; cur=cur->next){
    cout << cur->info << " ";
}
cout << endl;
for(node* cur = tail; cur!=NULL; cur=cur->prev){
    cout << cur->info << " ";
}
cout << endl;
```

Ввод-обработка-вывод

```
for(node* cur = head; cur!=NULL; cur=cur->next){
    cur->info *= 2;
}
for(node* cur = head; cur!=NULL; cur=cur->next){
    cout << cur->info << " ";
}
cout << endl;
for(node* cur = tail; cur!=NULL; cur=cur->prev){
    cout << cur->info << " ";
}
cout << endl;
```

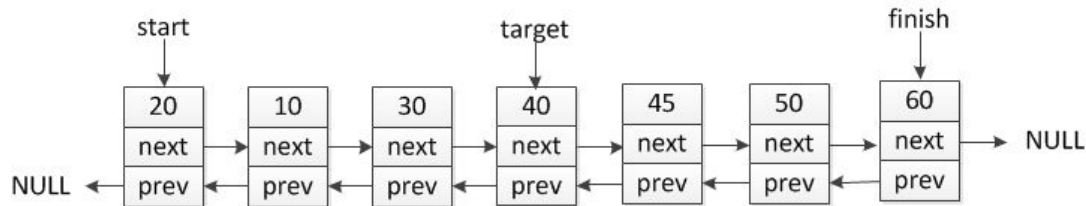

Вставка узла



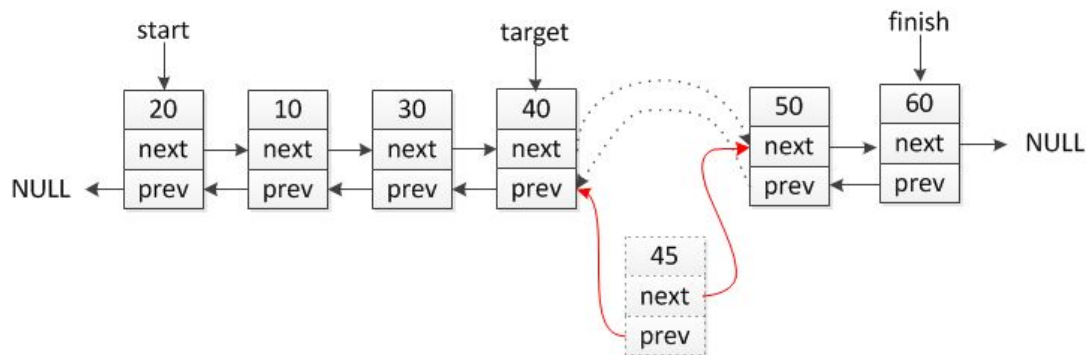
```
newNode = new node;  
newNode->info = 45;  
newNode->next = NULL;  
newNode->prev = NULL;
```

```
newNode->next = target->next;  
newNode->prev = target;  
target->next->prev = newNode;  
target->next = newNode;
```

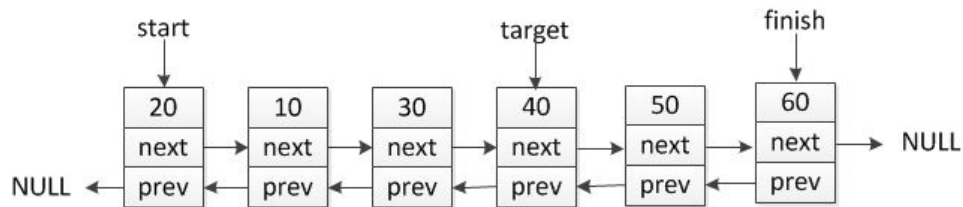
Удаление узла



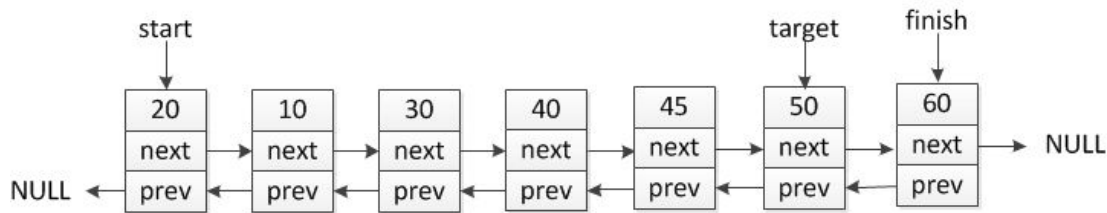
`killNode = target->next;`



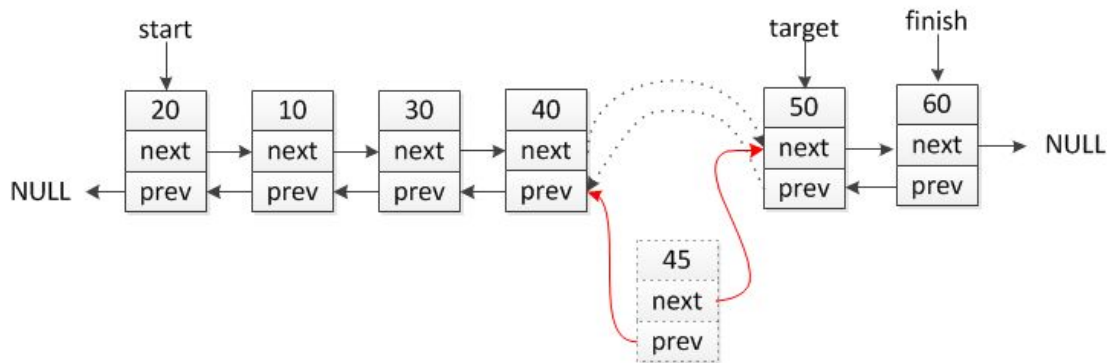
`target->next->next->prev = target;`
`target->next = target->next->next;`
`delete killNode;`



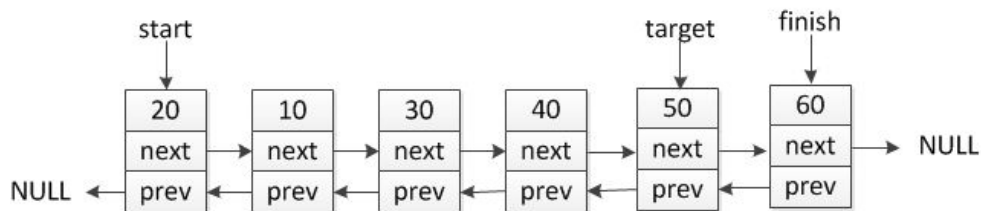
Удаление узла



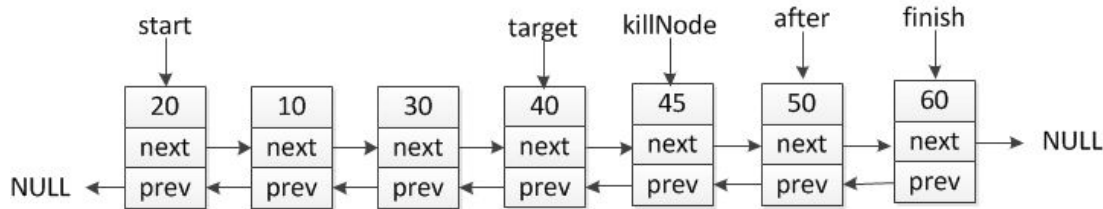
killNode = target->prev;



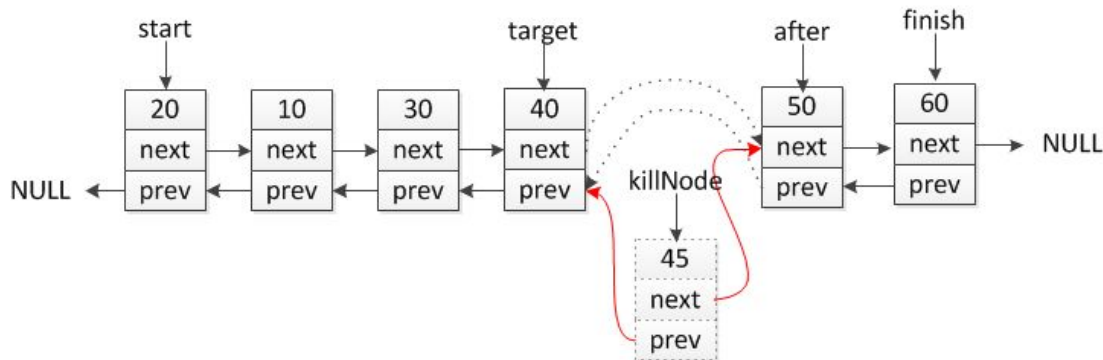
target->prev->prev->next = target;
target->prev = target->prev->prev;
delete killNode;



Удаление узла



killNode = target->next;
after = target->next->next;



after ->prev = target;
target->next = after;
delete killNode;

