

دانشگاه بین المللی امام خمینی



IMAM KHOMEINI
INTERNATIONAL UNIVERSITY

دانشگاه بین المللی امام خمینی قزوین

دانشکده فنی و مهندسی

مبانی کامپیوتر و برنامه سازی

(دومین ترم کرونا!)

فصل پانزدهم: مقدمه ای بر لیست پیوندی

نستوه طاهری جوان

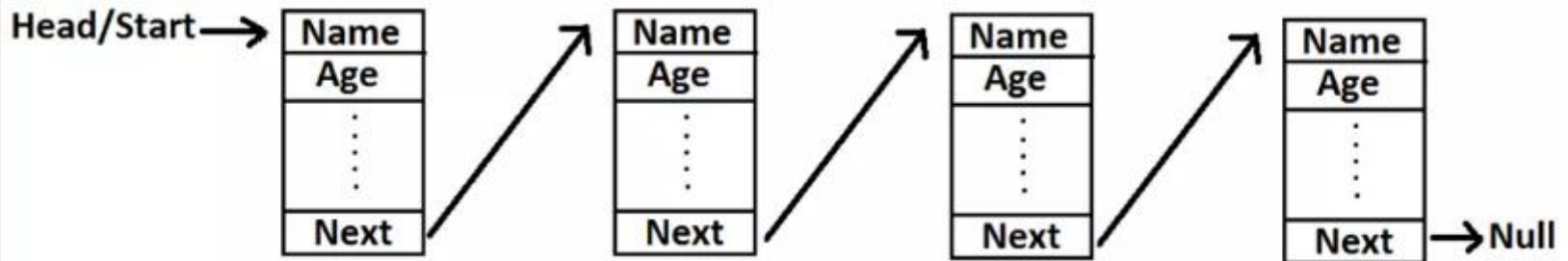
nastoooh@aut.ac.ir



مقدمه

✓ مفهوم لیست پیوندی

- یک لیست پیوندی از تعدادی گره تشکیل شده است.
- هر گره شامل مقداری داده، به همراه اتصال به گره بعدی است.
 - در واقع هر گره، آدرس گره بعدی را نگهداری می کند.
- گره ها ابتدا باید در حافظه ایجاد شوند، سپس لینک ها برقرار شود.



نمای شماتیک لیست پیوندی

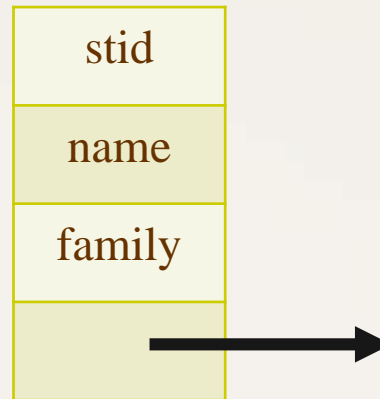


تعریف گره

✓ تعریف گره لیست پیوندی

- هر گره لیست پیوندی در واقع یک ساختمان است.
- هر گره شامل تعدادی فیلد داده ای به همراه یک فیلد آدرس (اشاره گر به گره بعدی) است.
- مثال: یک گره لیست پیوندی می تواند به صورت زیر باشد:

```
struct node{
    int std;
    char name[20];
    char family[20];
    struct node *link;
};
```





ایجاد گره

✓ ایجاد گره لیست پیوندی

○ برای ایجاد گره لیست پیوندی، از تخصیص پویای حافظه استفاده می کنیم.

- مثلاً با تابع `malloc()`.
- به تعداد گره می توانیم گره ایجاد کنیم.
- فقط باید مراقب اتصال ها باشیم.

○ مثال:

```
struct node *p;
```

```
p = malloc ( sizeof (struct node) );
```

- با اجرای این دستور، جایی از حافظه (کجا؟ مهم است؟) به اندازه ساختمان ما تخصیص حافظه شده و آدرس آن در اشاره گر `p` قرار می گیرد.



پیوند گره ها

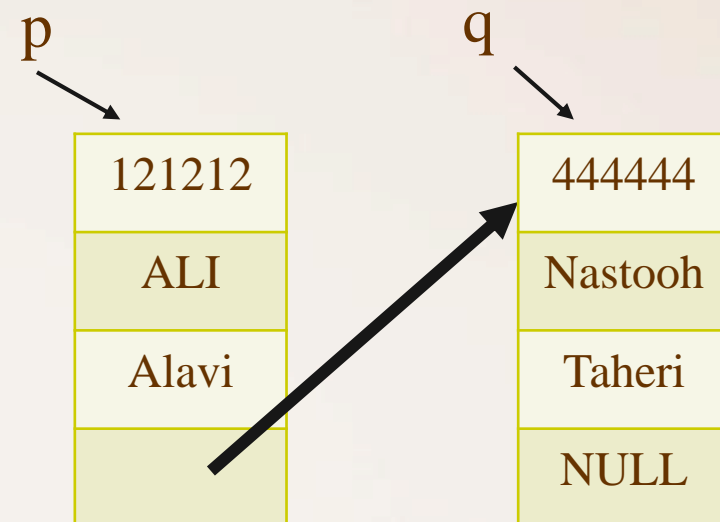
✓ پیوند دادن گره‌های لیست پیوندی

○ مثال:

```

struct node *p, *q;
p = malloc ( sizeof (struct node) );
q = malloc ( sizeof (struct node) );
p-stid = 121212;
strcpy(p->name, "ALI");
strcpy(p->family, "Alavi");
q-stid = 444444;
strcpy(q->name, "NastooH");
strcpy(q->family, "Taheri");
p->link = q;
q->link = NULL;

```



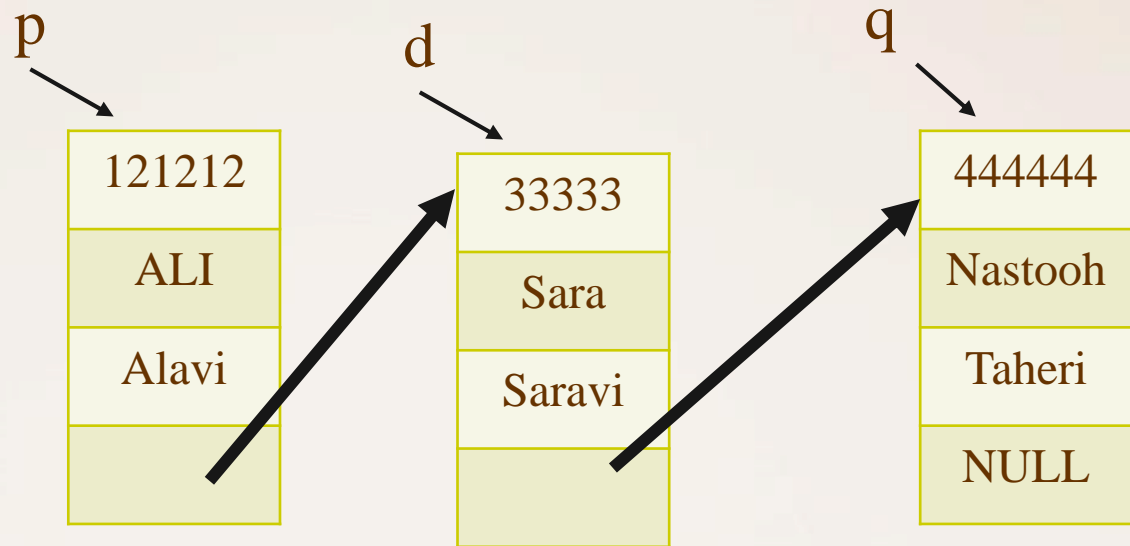


افزودن گره

✓ اضافه کردن گره به لیست پیوندی

- فرض کنیم گره d را بخواهیم میان p و q درج کنیم.
- پس از ایجاد گره d و مقدار دهی فیلدهای داده ای آن:

$p \rightarrow \text{link} = d;$
 $d \rightarrow \text{link} = q;$



بعد از اتمام کار عملاً به d دیگر نیازی نداریم. به q چطور؟



افزودن گره

✓ اضافه کردن گره به لیست پیوندی

- فرض کنیم بخواهیم به ابتدای لیست قبلی یک گره اضافه کنیم.
- می توانیم از همان اشاره گر d استفاده کنیم.

```
d = malloc( sizeof(struct node));
```

```
d->link = p;
```

```
p = d;
```

برای اینکه کماکان
به ابتدای لیست
اشاره کند.





افزودن گره

✓ اضافه کردن گره به لیست پیوندی

- فرض کنیم بخواهیم به انتهای لیست قبلی یک گره اضافه کنیم.
- می توانیم از همان اشاره گر d استفاده کنیم.

```
d = malloc( sizeof(struct node));
```

```
q->link = d;
```

```
q = d;
```

برای اینکه کماکان
q به انتهای لیست
اشاره کند.





اشاره گرها

✓ به چند اشاره گر نیاز داریم؟

○ معمولا برای کار با یک لیست پیوندی (حتی طولانی) به ۲-۳ اشاره گر نیاز داریم.

- یک اشاره گر به اولین گره لیست پیوندی اشاره می کند.
 - مانند p در مثال های پیشین. (بهتر است نام آن را start انتخاب کنیم!)
- برای پیمایش و اعمال تغییرات بر روی لیست پیوندی به اشاره گرهای کمکی نیاز داریم.
 - مانند d در مثال های پیشین.
- گاهی یک اشاره گر نیز به گره انتهایی اشاره می کند که عمل افزودن گره به انتهای لیست را ساده تر می کند.
 - مانند q در مثال های پیشین. (بهتر از نام آن را end یا مشابه آن انتخاب کنیم)



پیمایش لیست

✓ پیمایش لیست پیوندی

- گاهی نیاز است برای جستجوی داده ای خاص، لیست پیوندی پیمایش شود.
- مثال: چاپ اسامی تمام افراد موجود در لیست:
 - فرض میکنیم لیست پیوندی با تعداد زیادی گره وجود دارد.
 - فقط باید آدرس اولین گره در اشاره گر start وجود داشته باشد.
 - معمولا به start دست نمیزنیم، بلکه از یک اشاره گر کمکی استفاده می کنیم.

```
temp = start;  
do{  
    puts (temp->name);  
    temp = temp->link;  
}while (temp != NULL);
```



پیمایش لیست

✓ پیمایش لیست پیوندی

○ مثال: چاپ اسم فردی با شماره دانشجویی ۵۵۵۵:

- فرض میکنیم لیست پیوندی با تعداد زیادی گره وجود دارد.
- فقط باید آدرس اولین گره در اشاره گر start وجود داشته باشد.
- معمولا به start دست نمیزنیم، بلکه از یک اشاره گر کمکی استفاده می کنیم.

```
temp = start;
while (temp != NULL)
{
    if (temp -> stid == 5555)
        puts (temp->name);
    temp = temp->link;
}
```

تمام لیست پیمایش می شود، به امید اینکه چند نفر با شماره ۵۵۵۵ وجود داشته باشند!

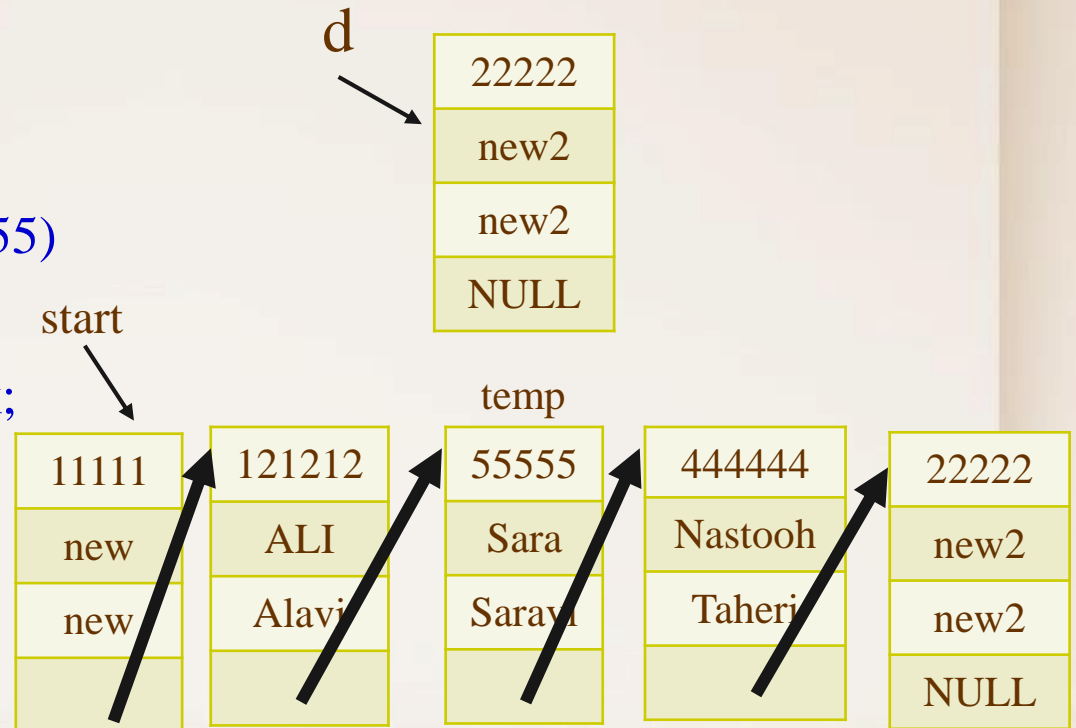


پیمایش لیست

✓ پیمایش لیست پیوندی

- مثال: افزودن یک گره پس از گره ای با شماره دانشجویی ۵۵۵۵:
- فرض میکنیم مشخصات گره جدید در `d` وجود دارد.

```
temp = start;
while (temp != NULL)
{
    if (temp -> std == 5555)
    {
        d->link = temp->link;
        temp -> link = d;
    }
    temp = temp->link;
}
```

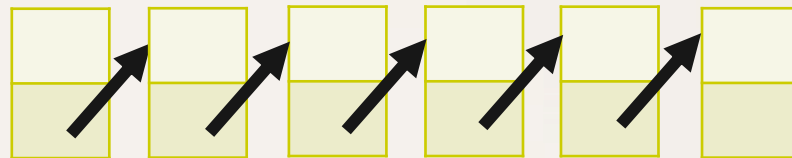




?

✓ سوال: قطعه کد زیر با چه هدفی نوشته شده است؟

```
temp = n = start;
while (temp != NULL)
{
    if (temp -> stid == 5555)
    {
        n->link = temp->link;
        free(temp);
        break;
    }
    n = temp;
    temp = temp->link.
}
```





✓ چند نکته برای کار با لیست پیوندی

○ هنگام کار با لیست پیوندی باید استثناهایی مانند زیر را در نظر گرفت:

- ممکن است لیست خالی باشد.
- مثلا هنگام افزودن یا حذف.
- ممکن است لیست فقط شامل یک گره باشد.
- مثلا موقع حذف همان یک گره.

○ لیست های پیوندی می توانند حلقوی باشند.

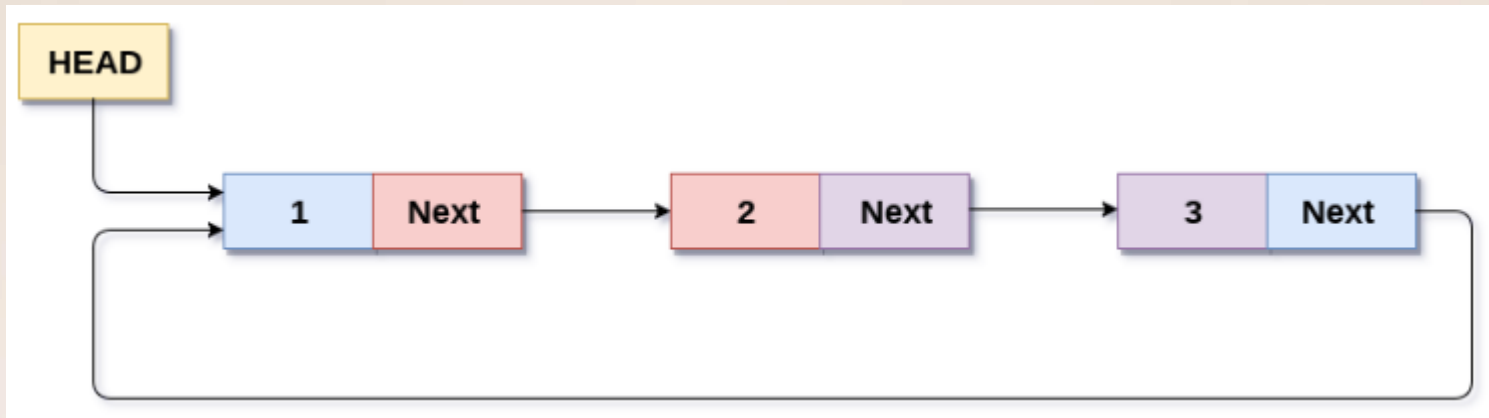
- یعنی گره آخر به گره اول اشاره کند.

○ لیست های پیوندی می توانند دو پیوندی باشند.

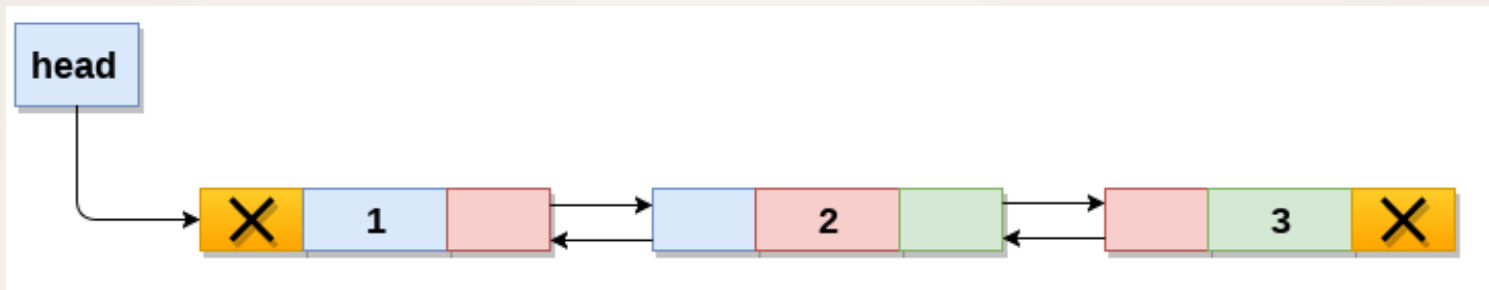
- یعنی هر گره یک اشاره گر به گره بعدی و یک اشاره گر به گره قبلی داشته باشد.



✓ چند نکته برای کار با لیست پیوندی



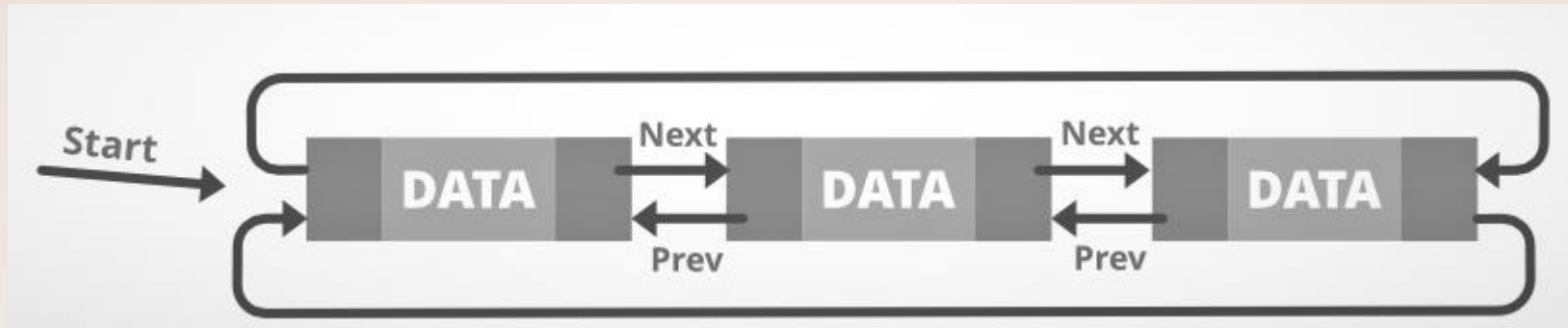
لیست حلقوی



لیست دو پیوندی



✓ چند نکته برای کار با لیست پیوندی



لیست حلقوی دو پیوندی



پایان