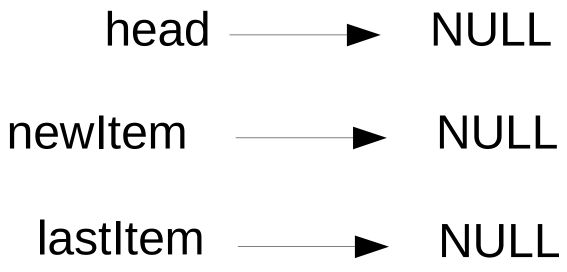
Lista Encadeada em Assembly

Prof. Ronaldo Luiz Alonso

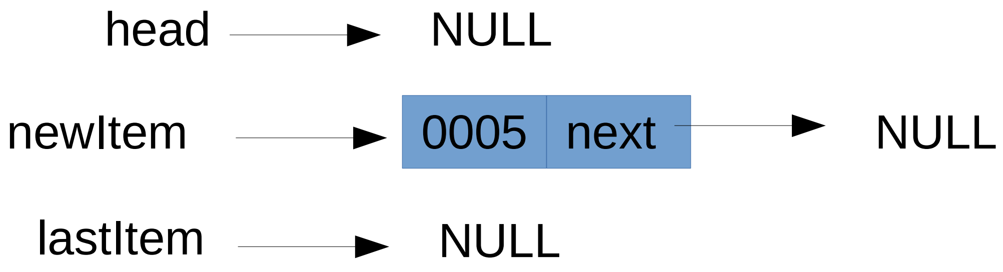
Ciência da Computação - UFMT

Estrutura de Dados

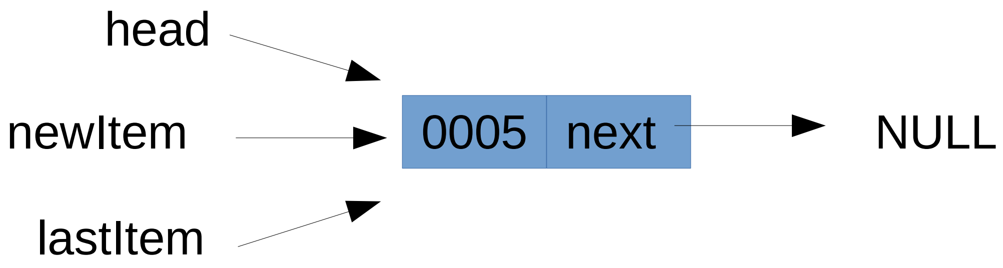
```
struc   linklist
      next:  resd   1
      nr:     resd   1
endstruc

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

%define NULL           0
%define LLSIZE          8
```



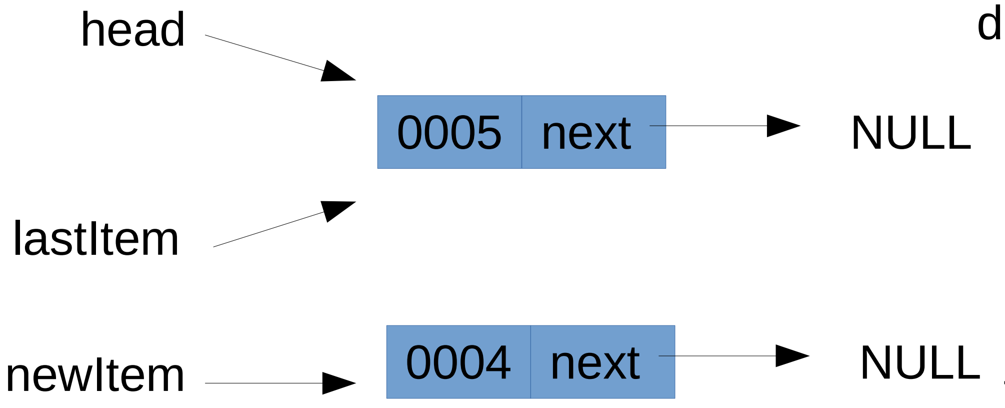
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



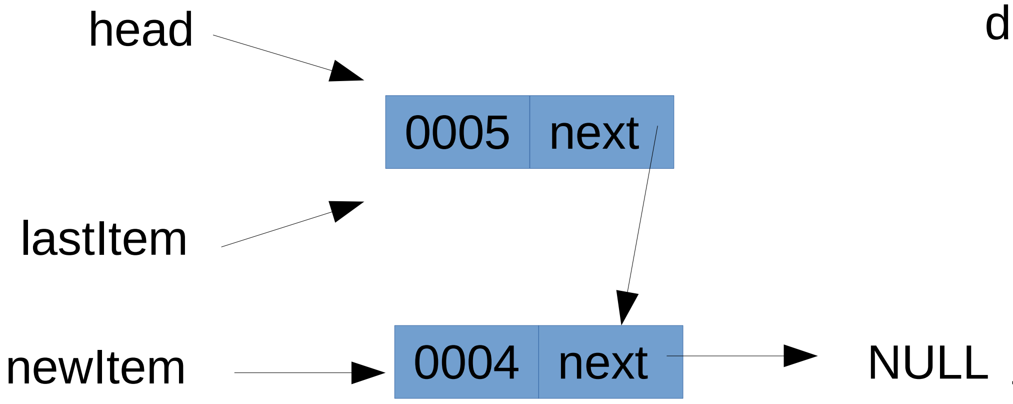
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



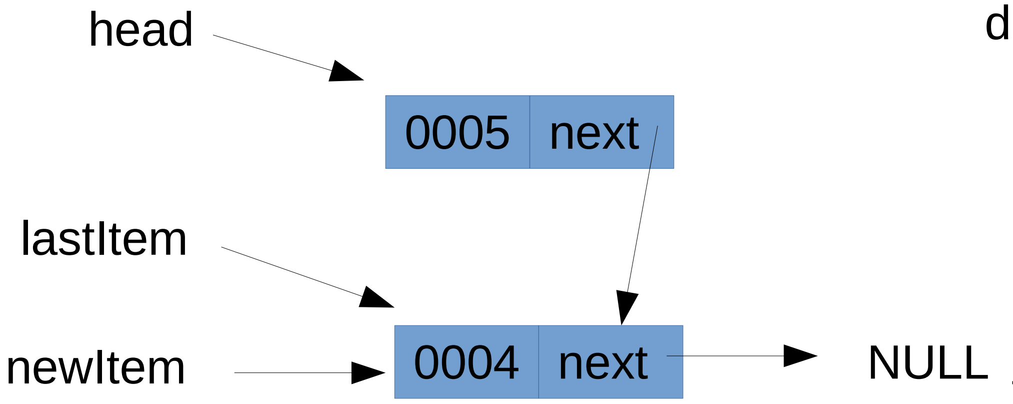
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



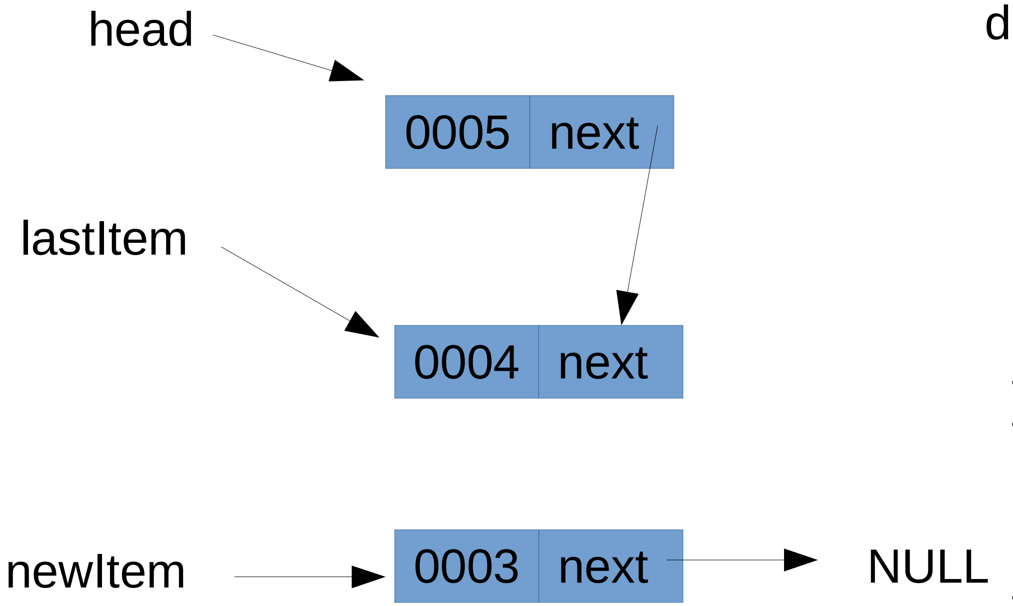
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



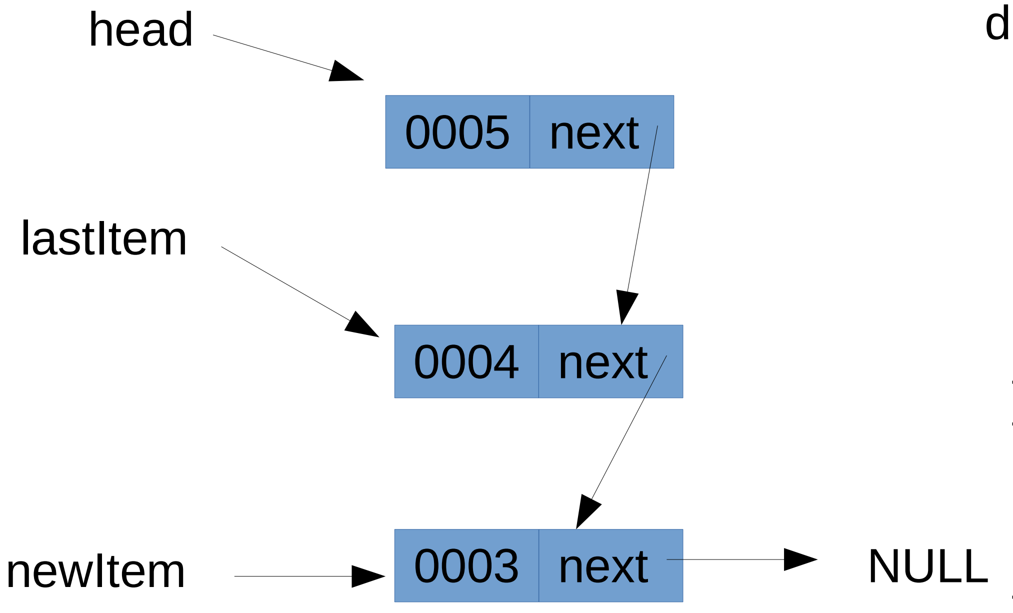
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



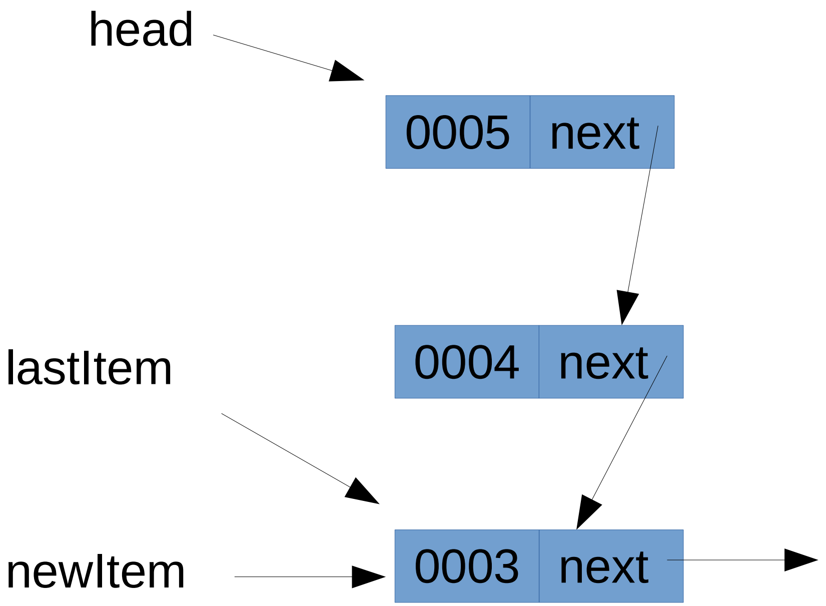
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



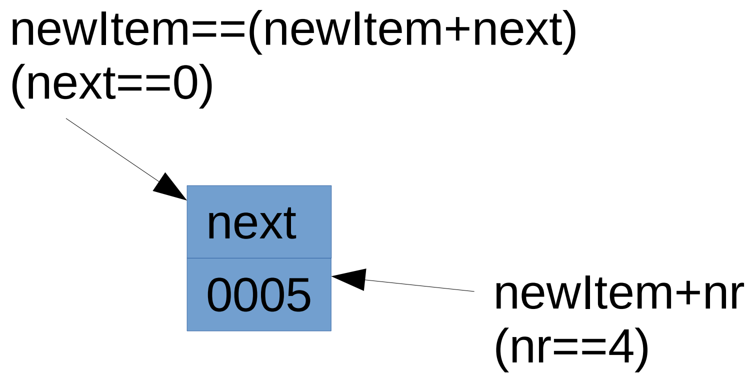
```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```

```
limit  dd 5
main:
        mov     dword [ head ], NULL
        mov     dword [ newItem ], NULL
        mov     dword [lastItem ], NULL


do_while_1:

        mov     rdi, LLSIZE
        call    malloc
        mov     dword [newItem], eax

        mov     ecx, dword [limit]

        mov     ebx, eax
        mov     [ ebx ], dword NULL
        mov     [ ebx + nr ], ecx
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```

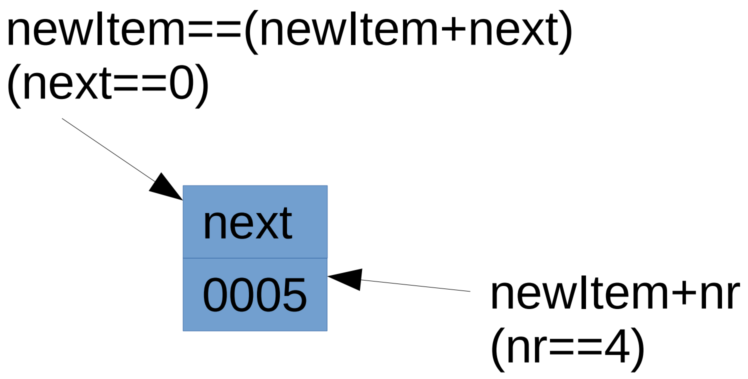
```
limit  dd 5
main:
        mov     dword [ head ], NULL
        mov     dword [ newItem ], NULL
        mov     dword [lastItem ], NULL


do_while_1:

        mov     rdi, LLSIZE
        call    malloc
        mov     dword [newItem], eax

        mov     ecx, dword [limit]

        mov     ebx, eax
        mov     [ ebx ], dword NULL
        mov     [ ebx + nr ], ecx
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



```
do_while_1:

        mov     rdi, LLSIZE
        call    malloc
        mov     dword [newItem], eax

        mov     ecx, dword [limit]

        mov     ebx, eax
        mov     [ ebx ], dword NULL
        mov     [ ebx + nr ], ecx
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```



```
do_while_1:

        mov     rdi, LLSIZE
        call    malloc
        mov     dword [newItem], eax

        mov     ecx, dword [limit]

        mov     ebx, eax
        mov     [ ebx ], dword NULL
        mov     [ ebx + nr ], ecx
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```

```
limit  dd 5
main:
        mov     dword [ head ], NULL
        mov     dword [ newItem ], NULL
        mov     dword [lastItem ], NULL


do_while_1:

        mov     rdi, LLSIZE
        call    malloc
        mov     dword [newItem], eax

        mov     ecx, dword [limit]

        mov     ebx, eax
        mov     [ ebx ], dword NULL
        mov     [ ebx + nr ], ecx
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```

```
      cmp     dword [ head ], NULL
       jne     parte_else
       mov      [ head ], eax
       mov      [lastItem], eax
       jmp     fim_if

parte_else:
       mov   ebx, dword [lastItem]
       mov   edx, [newItem]
       mov    [ebx], edx
       mov   dword [lastItem], edx
fim_if:
       mov     ecx, dword [limit]
       dec ecx
       mov    dword [limit],ecx
       cmp  ecx,0
       jne do_while_1
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```

```
      cmp     dword [ head ], NULL
       jne     parte_else
       mov      [ head ], eax
       mov      [lastItem], eax
       jmp     fim_if

parte_else:
       mov   ebx, dword [lastItem]
       mov   edx, [newItem]
       mov    [ebx], edx
       mov   dword [lastItem], edx
fim_if:
       mov      ecx, dword [limit]
       dec ecx
       mov    dword [limit],ecx
       cmp  ecx,0
       jne do_while_1
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```

```
       cmp     dword [ head ], NULL
        jne     parte_else
        mov      [ head ], eax
        mov      [lastItem], eax
        jmp     fim_if

parte_else:
        mov   ebx, dword [lastItem]
        mov   edx, [newItem]
        mov    [ebx], edx
        mov   dword [lastItem], edx
fim_if:
        mov     ecx, dword [limit]
        dec ecx
        mov    dword [limit],ecx
        cmp  ecx,0
        jne do_while_1
```

```
limit = 5;
  head = NULL;
  newItem = NULL;
  lastItem = NULL;
 // cria a lista
do {
  newItem = malloc (sizeof(struct linklist));
  newItem-> next = NULL;
  newItem -> nr = limit;
  if ( head == NULL) {
      head = newItem;
      lastItem = head;
  }
  else {
    lastItem -> next = newItem;
    lastItem = newItem;
  }
  limit --;

}while (limit > 0);
```