

Announcements

- Quiz 1 out today! Due Sunday at 11:30 PM.
- Tentative Schedule

Quick Note

```
struct node {  
    int data;
```

⇒ self-referential structures

```
    struct node * next;  
};
```

→ a member of this structure points to another struct of the type.

```
struct node {  
    int data;  
    struct node next;  
};
```

→ What is happening here?

struct node * insert_at_end (struct node * head, int data)
 using double pointers!

void insert_at_end (struct node ** p_head, int data)

Code

Stack

Heap

```

struct node *
runner = *p_head;
(value at 0x5000)
Struct node * newNode
= NULL;
newNode->data = data;
newNode->next = NULL;
newNode = (struct
node *)
malloc (sizeof
(struct node))
    
```

insert_at_end()

newNode 0x400

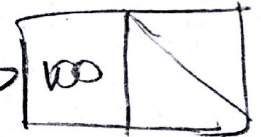
runner 0x100

phead 0x5000

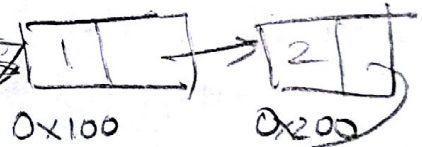
Main()

head 0x100

0x5000



0x400



0x100

0x200

0x300

Main()

```

struct node * head
= NULL;
head = buildList();
insert_at_end(&head, 100)
    
```

Case ① List is Empty

```

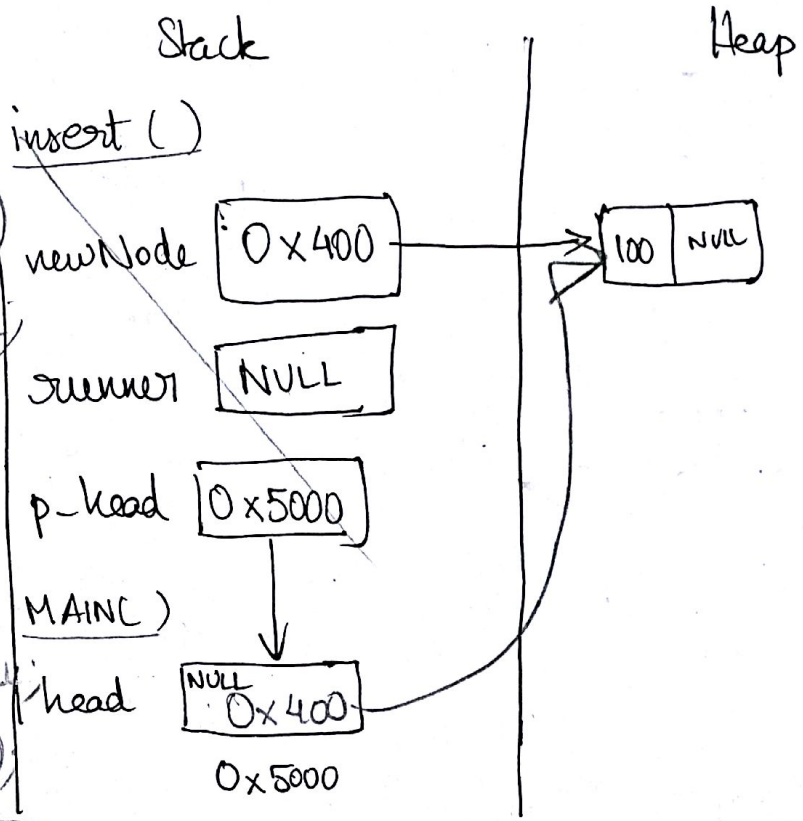
if (sumner == NULL)
    *p_head = newNode;
return;
}

```

```

main() {
    struct node * head = NULL;
    insert_at_end(&head, 100);
    // List of length 1
}

```

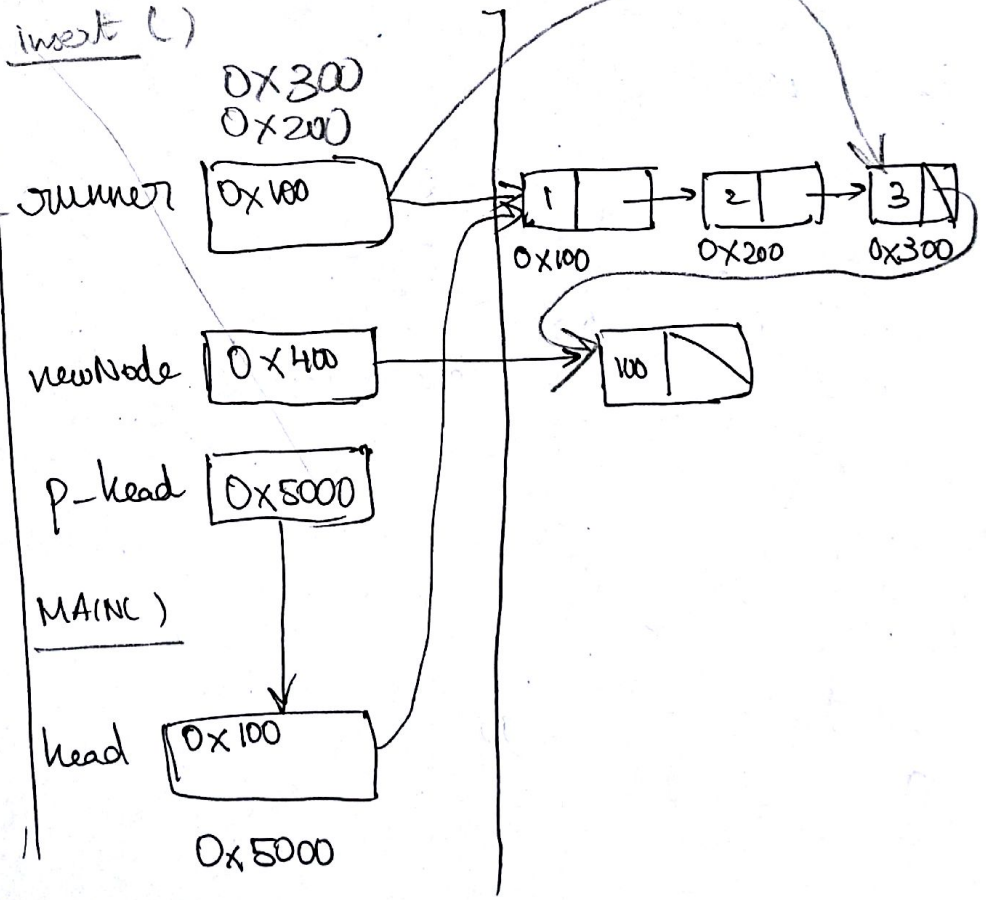


Case ② List is Not Empty

```

while (sumner -> next
    != NULL)
{
    sumner = sumner
    -> next;
}
// sumner will point to
the last node
sumner -> next
= newNode;
return; //
}

```

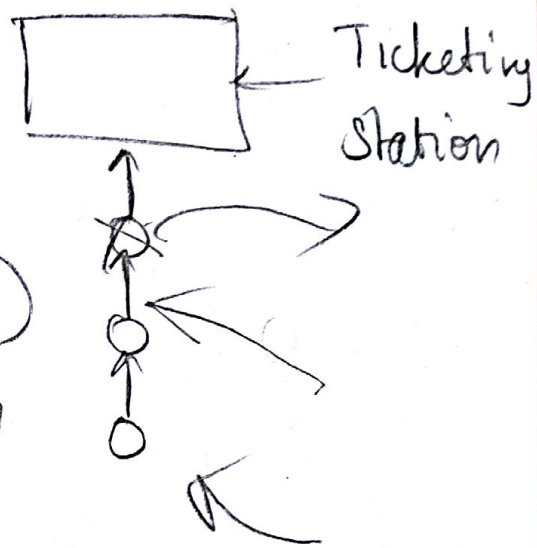


Queue

Controlled data access and insertion.

enqueue \rightarrow Add at the end (insert-at-end)

dequeue \rightarrow Remove node from the beginning (POP).



head

enqueue (x head, 1)

head

enqueue (x head, 2)

head

dequeue (x head)

head

Preprocessor directives → # define
→ # include

define directive

Magic numbers are bad.

Use macro substitution

→ A fragment of code
that has been given a
name.

define → ends with a
new line

include

Imports the header files and adds it before the code you have in your current file

include <stdio.h> → searches the systems library folders.

include "myheader.h" → searches your local directory.

```
int main ( )
```

```
{
```

```
printf ( " %d \n " , var );
```

```
// 1998
```

```
sayHello ( );
```

```
return;
```

```
}
```

```
void sayHello ( )
```

```
{
```

```
printf ( " Hello " );
```

```
}
```

```
int var = 1998;  
void sayHello ( );
```

```
myheader.h
```

include_func.c

```
#include <stdio.h>

void sayHello ()
{
    printf ("Hello");
}
```

include_func.h

```
void sayHello ();
```

include_main.c

```
#include "includefunc.h"

int main ()
{
    sayHello ();
    return 0;
}
```

gcc -c include_main.c
gcc -c include_func.c

gcc include_main.o
include_func.o
-o include_main

./include_main

global variables
function prototypes
(declarations)

Typically declared
in headers

struct