02-09-2024

Acts@ Batch #2024
Harish@cdac09

Programming Fundamentals for Design & Verification

- Kishwar Sayed

- Keywords, Identifier.

- Introduction to C. Arrays, function, strings, structure & union
- Introduction to C++ Class & Object, Inheritance, Class & Template function

Exception handling & namespace.

%d, %f, %s, ← Format specifier

macro ->, short circuit evaluation && -> false
|| -> true.

03-09-2024

Prime ->

6 % 2 == 0 ✓ => not prime

Functions → declaration — return-type fn_name (parameter);
→ definition — return-type fn_name (parameter) { }
→ call — fn_name (values);

calling fn, called fn

Diff between -> call by value
call by address (not referred)

actual parameter, formal parameter } -> can have same name

multiple return statement

Pointer
* De-referencing operator
& address operator.

Execution stack LIFO
push, pop

05-09-2024 a[i] same as *(a+i)

Arrays

[] -> subscript operator

String -> \0 - null char
-> gets(str);

04-09-2024

gcc
- Debugger -g, Recursion.
(calling fn = called fn)

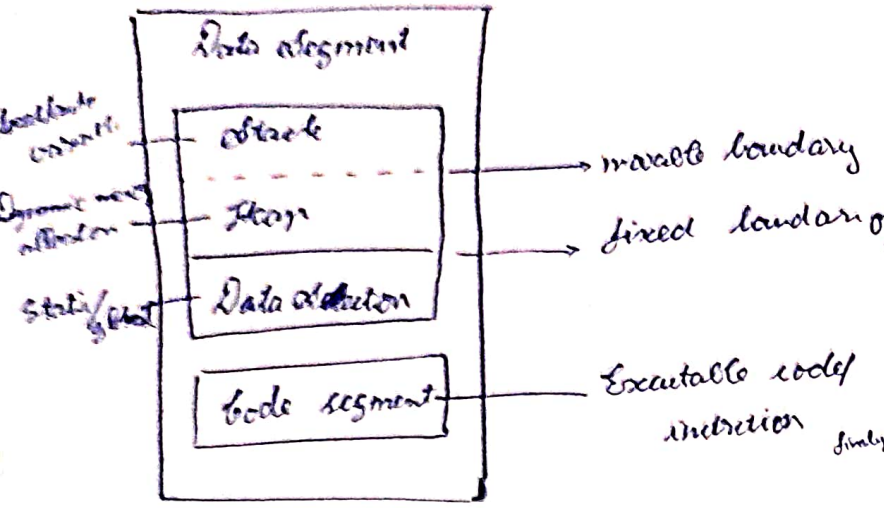
Swap

Pre-processor (-E)
macro - #define
file inclusion
macro
fn macro
conditional compilation
-> # all

06-09-2024
-> 2D arrays
memory allocated linear (contiguous)
-> line of code can increase - as compared to fn
a[1][2] same as *(*(a+1)+2)

recursion | Storage class

Memory diagram



Storage class

		location	init
auto	-	stack	garbage_value
extern	}	data section	0
static			0
register	-	CPU register	garbage val

07/09/2024

Dynamic memory allocation

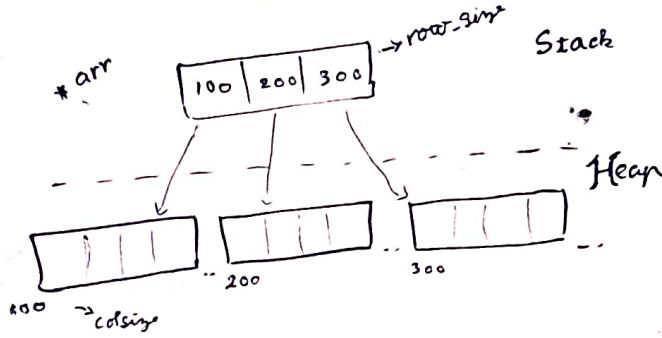
```
int *arr;
arr = (int *)malloc(size * sizeof (int));
        ↓
        calloc (size, sizeof (int));
        (calloc to '0')
free (arr);
realloc (arr, (n+num size) * size(int
        ↓
        (to change size)
```

finally ↳

Dynamic memory allocation for 2D array

With known (fixed) row & unknown (variable) column can be allocated in different memory as set of columns with known row size & address of each can be stored in pointer array.

```
int * arr[3];
```



09-09-2024 10-09-2024

Structures & Union → only one member can be used as shared memory by all

typedef → in using pointer

```
enum <tag_name> {m1, m2, .. mN};
enum days_of_week {Sun, Mon, Tue, Wed, Thu, Fri, Sat};
enum days_of_week day;
day = 'Wed';
```

11-09-2024 File Handeling

read, write, append, close

rb wb ab → BIN

a+ → read/write

```
rewind (*);
long ftell (*)
fseek (*, length, SEEK_END, SEEK_SET, SEER_CUR)
```

```
elipse <stdgrg.h> va_list (
(int a, ...)
```

12-09-2024

Object oriented programming → Abstraction, Encapsulation, Inheritance, Polymorphism

namespace (in) C++

reference variable

const →

inline function

function overloading

iostream → Cin, Cout } objects

>> Extraction operator

<< Insertion operator

1 — initialize array

for (loop 5 times iterate through each element in array)

2 — {

get input from user in array

}

3 — for (loop 5 times , iterate through each element in array)

{

display value of array at the index

}

length of str

for (i=0 ; str1[i] = '\0' ; i++)

{

}

print( )

str copy

for (i=0 ; str1[i] != '\0' ; i++)

{

str1[i] = str2[i]

}

str1[i] = '\0' ;

reverse -> str.cmp

str concatenate

str1 = str1 + str2

for (str.length = 0 , str1[i] != '\0' ; str.length ++);

for (i=0 , str2[i] = '\0' ; i++)

{

str1[i+str1.len] = str2[i] ;

}

str2[i] = '\0' ;

A - 65

- Documentation, W3
- data type, list, dict
- function
- search, start & display
- regress
- file - handling
- RPi cpu, gpiozero
- tensor flow, pyland
- Dylango,

Day 1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Day 2

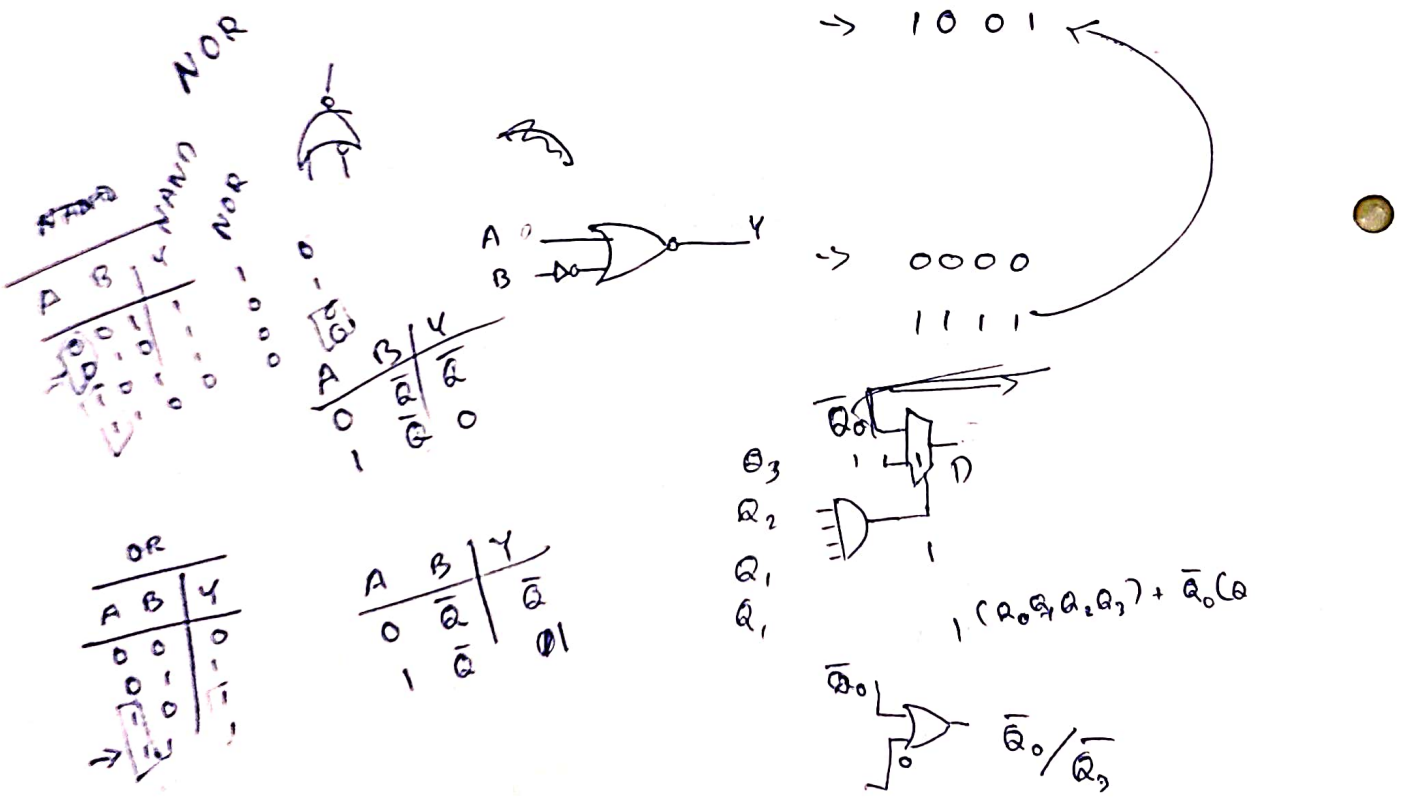
1.
2.
3.
4
5
6.
7.
- pointer
1.
2.
3.
- Extra
1
2.

Day 3

1
2.
3
4.
5.
- pre-process
1
2.
3
4.
5.
6.
7.

Day 4

1.
2.
3.
4.
5.
6.
- String
1.
2.
3.
4.
5.



```
flushall()
fflush(stdin)
while(getchar()!='\n');
```

18-09-2024

Linux Shell Scripting & Python

Kernel → manages resources

BIOS → MBR → GRUB → Kernel
(bootloader)

RAM tabs Swap (backing store)
Virtual memory

Run levels
0 → system halted
1 → single user mode
2 → multiuser mode without network.
3 → 2 + networking
4 → unused
5 → 3 + GUI
6 → Restart

In Terminal
$ - normal user.
# - root user
~ - home-dir
Linux commands

man
→ no. man for DOS commands.

23-09-2024
- user_add
→ /etc/passwd ⇒ 1 line/user
→ /etc/shadow
- new user gets added at end

Uid - 0-999 → system specific user
-1000+ → normal user
→ where is (rin)
→

group - group of user with similar permision

nl
wc -w
diff
chmod
chown
pc

⇒ Home not gets created in Ubuntu

26/09/2024
sh → shell /bin/sh
csh → C program /bin/csh
bash → Bourn again shell /bin/bash

Permisions
rwx --- ---
owner group other

→ /etc/group
⇒ group with same name gets created when add user.

usermod
/etc/skel → Skeleton.
.bashrc → run command

#!/bin/bash

'capital' ⇒ global variable (export) for application to use
'small' ⇒ local variable

read
echo

if - then fi
while
for - - done
-f -d
file directory.

tt -eq
-gt

number
@ individual [list]
* individual [string]

((i++))
Brace expansion {a...z}
a{b,c}d
abd acd
Str = 123456
range string.

Pg34-36

~~305, 309~~
269-278
ShellscriptingBible

27-09-2024
? - exit status
$ - process id

set -- □ □ □
parameter

check file permission.

grep -V
-i
-R

wc -w
-l
-c

ip addr
ip link

df
(top)

3 7 11 15 [redacted] → w → r [redacted]

Python

List = [ ]
- different datatypes

tuple = ( ) immutable → can add to new

set = set( ) { } → mutable

$\boxed{\text{Splicing}}$

File handeling, with :
fopen( )
close( )
write( )
read( )

Regex

re.search(r' ')
list ⇐ findall('pt', <list>)
list ⇐ split('pt', string)
split (, [ ]) ↳ no of occurance.
sub (" ", " ", <string) ↲ sum
org to be

Function

Definition. default parameter.
(named parameter), positional parameter
keyword parameter)

Class

init
super()
formated print

⇒

```
class Parent(object):
    def __init__(self, ...):
        self.... = ..
class Child (Parent):
    parent.__init__ = (        )
    super().__init__ (...) # same as object creation
```

Python, shell scripting

1. Using switch case ~~display~~ accept 2 different parameter.
   1. file exits.
   2. directory exits
   3. count files in directory
   4. Invalid choice.

2 Class, Parent → child

def display (first, last) → of a [ ]
to
def count('a') → count ( )
↳ in list

def len
def revers