

Kernel is Core of OS.

માગશર વદ ૧ શાકે ૧૦ પોષ



- bash - Bourne again shell

① Monolithic Kernel :- ex:- unix / linux / windows

VFS :- Virtual File System

PM :- Process Management

Multi processing Service.

Multi threading Service.

MMU :- Memory management

↳ memory allocation



IPC :- inter process communication

- PIPE

- msg

NPS :- Network protocol Service.

② Microkernel

mac/OS / minix

- size less

- execution Speed



- E Preprocessor

- V Verboge

- S

links → static links

માગશર વદ ૪ શાકે ૧૩ પોષ

→ dynamic links



bss → block started by symbol.

data → initialized static & global variable.

initialization

initialize. Var
able

શાહી 4

જાન્યુઆરી ૨૦૨૧

DFS :- Root file System
ext-1 - pendrive

All file inside \Rightarrow bootup time.

Runtime Code

init \rightarrow

Recursive Extended FAT

start - routine \rightarrow main

bin. lib. temp user bin etc dev proc active device

dev device logical

5 peripherals [block D.P [bus dev] char D.P [bus dev]]

key board, mouse etc]

\Rightarrow plt - procedure link

જાન્યુઆરી ૨૦૨૧

int open('file path', int flags, modes)

operator

kernel data type

- some controllers are connected to CPU via bus
- controllers are used due to diff. in speed of CPU & peripherals
- high level utilities = software

ELF :- executable Linkable file.

\Rightarrow current file offset :- measurement that measures no. of byte from the beginning of file.

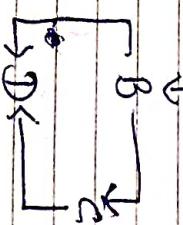
જાન્યુઆરી ૨૦૨૧

માગશર વડ ૫ શાહી ૧૪ પીએ

માગશર વડ ૧૧ શાહી ૧૬ પીએ

જાન્યુઆરી ૨૦૨૧

A



\rightarrow read (int file, void * , size_t n bytes)

\rightarrow write (- - -) .

\hookrightarrow return no. of written bytes

માગશર વડ ૧૨ શાહી ૨૦ પીએ

જાન્યુઆરી ૨૦૨૧

માગશર વડ ૫ શાહી ૧૪ પીએ

જાન્યુઆરી ૨૦૨૧

Q 11

Open file. write
2 buff [read empty buf
call write buf & write in
file & read & eat buf file

→ lseek system call performs repositioning of current file offset.

→ interpretation of lseek can see the current file offset

Q 12 SEEK_CUR (fd, 20, SEEK_U
SEEK-END)

→ if (REQ < MAX_ANU_SIZE):

→ read return less no. of byte then request make.

→ common API & System calls
↳ Application programming interface

Q 13 (a) 13 (b) 13 (c) 13 (d)

→ Harddrive.

types of blocks :-

- 1) Boot Block
- 2) Super blocks → structures info of RAM load, start, execute, terminate
- 3) inode blocks
- 4) data blocks

↳ file to structure file to structure permission, file to structure file to structure

d/9/2022

linear addressing of file
among multiple files.

call write buf & write in
file & read & eat buf file



→ dup is a system call a duplicate file system copy.

Q 15

→ interchangeable

→ syntax :- int dup(int fd);

→ dup2 for own fd number

int dup2(int fd, int fd2);

→ # include <fcntl.h> (optional)

fcntl(fd, F_DUPFD, 40);

Q 16

Q 14

Q 15

② process

3 process management

- program is in execution

RAM load, start, execute, terminate

- Linux can run multiple programs

- OS process concurrency

- makes to perform multitasking

- Copy on Write
- parent blocks act like page segments
- one copy of the child process added to each
- more efficient once host processes value Justice come for parent & child

fork

→ copy of main program

parent pid child pid)

return value;

child pid);

→ process termination:

- main return 0; } normal
- exit(0); } abnormal
- exit —> absolute

termination

→ execute C "main() of program"

fork

return value;

status

fork

return

value;

fork

8

atexit();
↳ deregistered functions (function pointers
are often declared as extern). deregistered
functions with kernel to execute
status.

On normal process termination
execute the fun in the reverse
order of their registration.

9

processes can register
multiple function pointers.

Syntax : atexit(void (*fun_ptr)(void))

atexit : atexit(Void (*fun-Ptr)(Void))

→ exit from calling the register
fun
by atexit fun & execute
in the reverse order of their
registration

→ -exit(c) : fun is not calling the
functions which are registered
by atexit fun

10

register by atexit fun
Very expensive in terms of CPU memory
usage

→ kernel processing

→ NT - Nice value
using which we can increase &
Priority of Process.

→ SZ - Size of Process
→ WCHAN - address in kernel space

where process is
sleeping.

पृष्ठ ३० ॥ १५ ॥ २२ अप्रै

11

→ TTY → terminal where process
is started.

→ pid_t wait(); #include <sys/wait.h>
address of exit status
pid_t waitpid(pid, int *stat, optional
options)

→ parent should calls wait system
call.

① Wait for Child block parent, if
child is still executing

② If child terminates immediately returns
wait status of terminated
child.

Execution error : when tried
to no child at all then it will
also you call wait(). then
it will show error.

→ disadvantage of fork :-

fork to maintain a process
executes lots of system calls

Very expensive in terms of
CPU memory usage

→ The ready : Creating, maintaining,
deregistering fast context
page to distinguish between
processes (in paged free)

→ threads are clearly not
processes (in paged free)

14

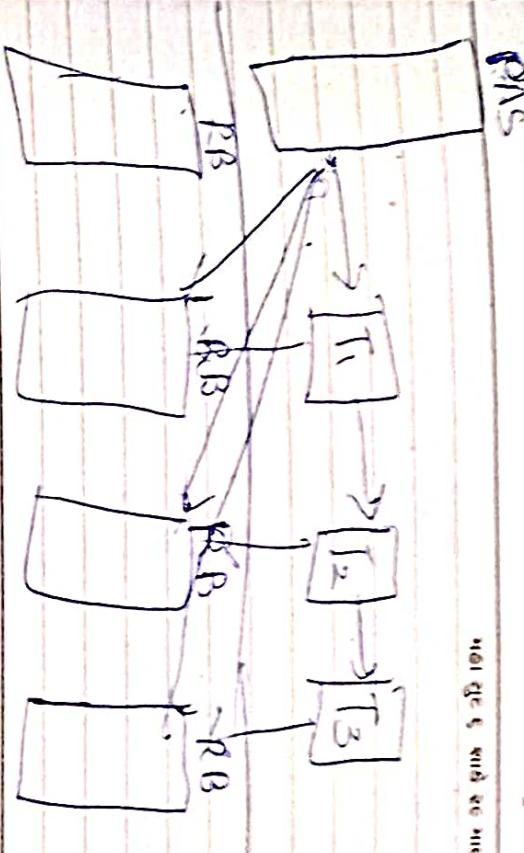
15

Thread is a thread context of execution.

- has one executing task.
- performs a specific task.

→ multiple threads in multiple user contexts of execution.

→ context switching is done simultaneously.

16

→ (i) of thread.

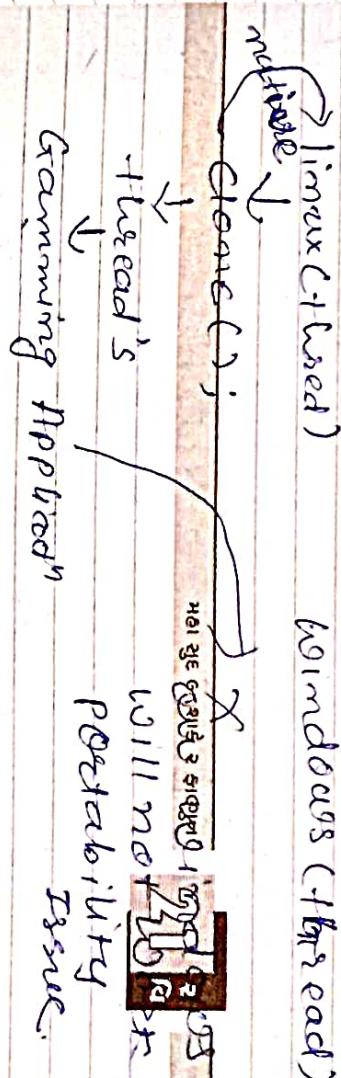
→ do not carry their own address space.

→ Share the address space at their threads.

→ common between threads and process happen by default.

→ (ii)

at least one thread is running a brief then switches to another scenario is switched.

17

→ threads -

- User level threads
- Kernel threads
- Processes

Windows (Process)

Portability issue

18

22

IEEE Launch POSIX Std.

25

Portable operating system interface
Set of functionality.

Appreciate s/ho.

possible en autre.

ପୋଷଣ
ବିଜ୍ଞାନ
ମାର୍ଗ ପତ୍ର
ଶାଖା ଓ କାଳେୟ

defining design of thread.
positive thread

~~POSIX Thread Creation~~

datatype

ପ୍ରକାଶକ
ମହାବିଦ୍ୟା
ପାଠ୍ୟ ପତ୍ର

pthread_t ⇒ Thread id Thread

Preload-att. \rightarrow Head att.

→ $y = -x^2 + 3$ spricht

^{int} P-hydroxy-Cresols C-P-hydroxy-C.

P. + L. + C. + O. - act. 2 * alt. 1

value. On the other hand, if you do not have enough permission, you will not be able to perform certain actions.

→ If you don't have enough resources, you will not be able to achieve your goal.

→ If you don't have enough time, you will not be able to complete your task.

→ If you don't have enough money, you will not be able to buy what you want.

→ If you don't have enough energy, you will not be able to work hard.

→ If you don't have enough knowledge, you will not be able to understand something.

→ If you don't have enough experience, you will not be able to handle certain situations.

→ If you don't have enough skills, you will not be able to perform certain tasks.

→ If you don't have enough motivation, you will not be able to stay focused and persistent.

→ If you don't have enough support, you will not be able to overcome challenges.

→ If you don't have enough resources, you will not be able to achieve your goal.

→ If you don't have enough time, you will not be able to complete your task.

→ If you don't have enough money, you will not be able to buy what you want.

→ If you don't have enough energy, you will not be able to work hard.

→ If you don't have enough knowledge, you will not be able to understand something.

→ If you don't have enough experience, you will not be able to handle certain situations.

→ If you don't have enough skills, you will not be able to perform certain tasks.

→ If you don't have enough motivation, you will not be able to stay focused and persistent.

→ If you don't have enough support, you will not be able to overcome challenges.

→ EPERM
don't have enough permission.

→ Joins given that did with many process execution.
Forces the process to wait for completion of threads.

① semaphores -
declaring the semaphore ^{object} available
sem - t . my - sem;

② Initialize Semaphores $\text{sem_init}(\text{C} \& \text{my_sem}, 1)$

is nonzero. address of initial condition

Semperore

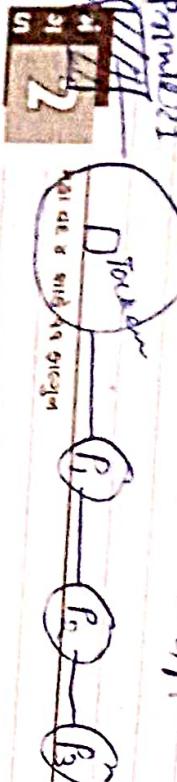
Sharing Some Photos
Sharing some photos between friends

QUESTION & ANSWER

(3) Sem-wait (Sem-ⁿ P1, P2)

- * Process operating on semaphore.
- ② Process take token from sem. opt.
- ③ Give back token to S/P

Semaphore token : 0, 1

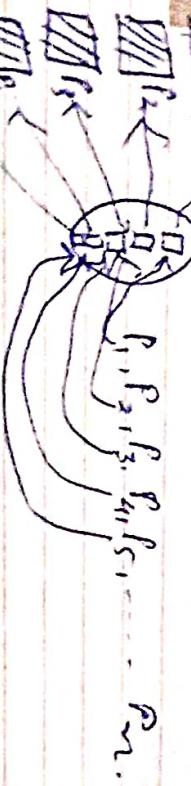


Types of Semaphore :-

- ① Binary Semaphore - A Semaphore which token value changes by 1 during the semaphore call.

② Counting Semaphore :-

3 P1, P2, P3, P4, P5, P6



③ Pthread-Spin lock (Pthread - Spinlock - t * P1, P2);

Other process can access circular process only perm.

5 3

④ Pthread-Spin-lock (Pthread - Spinlock - t * P1, P2);

- ① Pthread-Spin-lock (Pthread - Spinlock - t * P1, P2);

⑤ Context switching :-

- CPU switching of CPU from one executing process to another.

6 2

→ deadlock situation: where neither of the processes move farther.

→ other process releases the locks.

- deadlock avoidance processes to reduce this problem

A syn whose token count is variable
value > 2 is C/S.

15 श्रीमद्भागवत पुराण
प्राप्ति का अनुभव होता है। यह अनुभव एक विशेष घटना है जो लोगों के द्वारा अक्षय रूप से अनुभव किया जाता है। यह अनुभव एक विशेष घटना है जो लोगों के द्वारा अक्षय रूप से अनुभव किया जाता है।

Schedule → Schedule once & proceed once. Initialize a function which takes no arg. & return nothing.

~~1~~ Schiedskalar Attentante.

અંગેણ કુટુંબ શાહે ૨૫ ફાલ્યુન
16

Scheduling
Policy & Priorities

SPLIT THREAD INHERIT

→ ~~Patent~~ When a Product Changes
or Improves

A new + increased
percentage has still been set
at 10 percent
so scheduling it down.

→ There is a one member of an inheritance schedule member that specifies whether a trustee is inheriting the police & prioritizing from its parent or managing on its own

Francesca Parent ist gestorben.
Francesca Parent ist gestorben.
Francesca Parent ist gestorben.
Francesca Parent ist gestorben.
Francesca Parent ist gestorben.

\Rightarrow or if else, form PTFREP-EXPLICT-SCHEM. It has its own policy & Priority.

→ Spurteach = unter-Getrimm herz + schroff
C. & A. (s. L. imfo),

→ P + HCO₃ = CH₃COO⁻ + SO₄²⁻
CH₃COO⁻, R^{imgo},
CH₃COOH, Na₂SO₄ शाक व फल जैविक

• Linux thread priority & policies

Policy
Real time
Policies

	L	R	F T F O.
Default	Round	Round	1
all spread	- square	कागज युपि शर्करा कल्पना	
have same	- majority	- SCHIED-FIR	20 रुपये

Priority, - all get time
- same CPU incircular
no in

Slice time Sharing -SCHERZER

FÖRST DEPENDEO
AN ASYLIC SMC. 15 MED. OTHER

SCHEP-NORMMALE

22. Scheduling

Nice Value

Priority

Clock Rate

Time

Processor

Processor

- Preempting is done by changing priority of process.
- Process never leave CPU once interrupted.
- Priority value last is done.
- Round robin is used.

23

Priority

Priority

Priority

Priority

Priority

- Linux employs complete fair scheduling. (CFS)
- each & every process we will get proportion of CPU based on factors.
- nice value gives weightage to process with respect to CPU process time.
- can set the priority of process.

24

Priority

Priority

Priority

Priority

Priority

Priority

Priority

Priority

- command :- taskset
- it is assign a CPU core to a process.
- It is a utility tool comes by default with Linux distribution.
- Taskset -c 1 filename
- changing CPU core for process.
- Process has a nice value.
- Special symbol.
- Forces command to execute.

25

Priority

29
Date: 08/06/2024

A scheduling algorithms:

- ① O(n)
- ② FCFS - shortest job first
- ③ SJF - first in first out
- ④ RR - Round Robin

RR - Round Robin :-

30
Date: 08/06/2024

- each & every process are going to get equal proportion of CPU time.

- process has finite time slice.

CPU moves to the next process.

- each & every process will get same amount of CPU resources for same amount of time.

- Round robin is a form of preemptive scheduling process.

→ process can execute in

→ Process Context

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

Digitized by srujanika@gmail.com

→ Menschen (Vivid + active)

Désiré René

88

~~100~~ 100
~~100~~ 100

INTO WHICH THE

```
► memcmp Cvoid *addr, void *addr
```

B M P.
H C P.
P.
S
E
T
Y
R
I
N
E

- kernel executing a piece of context on behalf of a interrupt to user I.C.
- Linux MAD (Memory Management under).
→ memory manipulation calls:-

- > The moment process executes memory jumps to two add locations of start comparing address & part by byte until it gets summarized data, returning $4,0^{-1}$
- comparing data based on ASCII value & measure its difference:

```
for( i=0; i<n; i++ )  
    cout << arr[i] << " ";
```

~~→ Void * Memmove C void * dest, const void * src, size_t n;~~

四

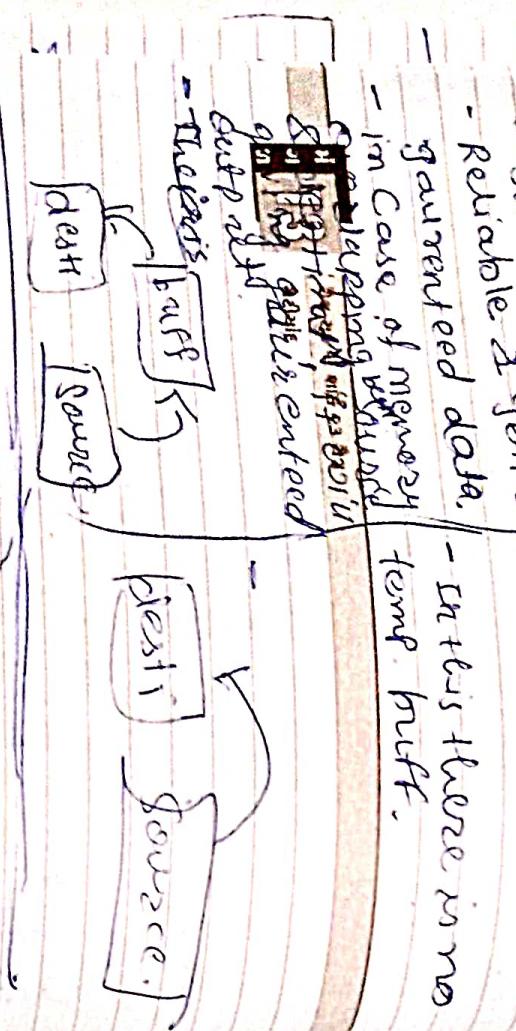
- When process executes `Memset` function jumps to the address location provided by the pointer argument & starts writing data with a given constant `c` for n no of bytes given by 3rd argument.

returning
derivative add. derivative
→ The moment ~~processes data buffers~~
mem move for copies data from
source buffer to dest. buffer
for given no. of bytes
dest. buffer gets contribution

12

\Rightarrow void * memcpy (void * dest, void * src, size_t size);

- Memmove
- Slow
- Reliable & general
- Guaranteed data.
- In this there is no temp. buff.



alloc:

- Void * malloc (size_t nbytes);

- The moment process executes malloc it allocates memory from stack segment if heap segment is full & return pointer to the region of success.

② repetitive use allocation -
overflow will take place.

Stack 1) lock:

int lock (void *addr, size_t nbytes);

15

- min. memory wastage
- copy faster than alloc.
- alloc doesn't maintain pool of fixed 100x sizes. hence no memory fragmentation
- needn't to call free memory call free memory automatically deallocate them funds terminates.

As swapping process:-

- When RAM is running out of memory & OS wants to launch new application RAM & OS looks for processes from the RAM & move them into the swap partition of your storage devices via Swap out process.

- When newly app done their job, processes remain will get back the processes from swap partition for RAM & via Swap in process.

* Memory locks:-

- lock could be a login program we don't want someone to log in my login program from the RAM (then we can apply memory locks).

→ Types of memory locks:

18

मे 2021

3

- frame no. & frame number in page no. in page table
- take the physical address of frame leading to physical address of file.
- last variable everet I/O request for each processes has to perform operation to physical address + an logical file address is taken. This memory address state is referred as translation.

4

- Virtual Memory, a section of memory mapped partition of storage device.
- Linux uses Virtual memory & virtual addresses assigned to your process.

5

- force command provide amount of virtual memory & need

- command line virtual memory & need to map it.

→ cat /proc/

- Vmstat = Virtual memory statistic - D.L. Creation uses a flag class that b - uninteruptable process. pi - blocks received from block device. User may should get ready to get

मे 2021

6

- block sent to block device
- no. of interrupt/sec.
- no. of context switch/sec.
- library :- It is a group of pre-compile object files.
- Library :- It is a group of pre-compile object files.

5

- static libraries are one which are statically link to program executable file at compile time.
- dynamic libraries are one which are dynamically link to program at runtime.
- Linux OS has a static library with extension a ex:- abc.o
- generate object file respectively.
- as is Linux cmd line tool like archive that is used to create static library.

- ccs (replace, create symbol)
- -L linking the library lib-st.a from current directory.

6

- dynamic library :-
- f-pic

- position independent code

- file extensions is .so
- they are also classed as shared libraries.
- D.L. Creation uses a flag class D.L. should be relocatable that means the instructions of dynamic

मे 2029

10

load into any memory or

- space & executable addresses of D.L. & J.
- need the addresses with the processor
- needs the memory that D.L. appear
- such a way that executable file
- be fast if executable file
- program another executable file instruction
- latter will be shared get message
- of memory is used to generate
- shared
- D.L. Id is a linux command too!
- since Id is dependency
- will pass the executable file required for executable file.

1000 bytes :-)

13

100

- once the kernel bootup starts
processor shifted to protected
mode & locking memory as a
set of block chunks of pages.
This process called process initiation.

- kernel OS लगावली प्रक्रिया संख्या (एड-डिस्ट्रिब्युशन)

- some illusion & making your

CPU to look memory as set of block

gdb GNU debugger tool:-

- gdb you can use source software

- is a command line tool.

- works with executable files that are

- produced by compilation process.

- it has debugger tool that

- supports C, C++, Fortran, Python, Pascal & Fortan etc.

- PROTECTED mode

- once the system has power or during booting of processor

- Linux made locking memory as array of bits

- provide the gdb & executable file
program execution and to the
value - you info locals - will display
value of all the local variable.

मे 2029

14

50

15

50

11

प्र० १०५ २०२८

प्र० १०५ २०२८

50

logical acid.
+ offset

Physical address
CfZ 3000-3000

2029

દૈશાળ સુંદર
શાહી અનુભૂતિ

20

$\rightarrow \$ \text{ gib } \cdot \text{RE}^* \cdot \text{arg1 arg2 arg3}$
 (gdb)

`→ $ gdb -i Ex
gdb) set arg1 arg2 arg3
gdb) run`

→ If you observe real time application
of the sample nest of the
leaving young happen to be
seen near point of view.

Ex.: desreferencing NUL pointers
→ - - - - - uninitialized pointers
→ - - - - - pointer which is already free.

→ accessing a memory beyond the lexical or semantic boundary

→ When process created, Process at a Space is allocated & Program is mapped.

Western Hemisphere program
makes application "Segment" fail →
answer

Of a structure provide please
provide min alignment for the
structure variable তথ্য কু ৭৭ শাক্ত ২ পরিষ

→ gdb fails to trap docn. ~~key~~ ^{key} 123

-CPU will get the logical address & map to respective physical addresses

frame Selection Table (K.P.F.R.T.)

CPV giving the same to K.P.R.
Kesmeli fails to keep frame no.
for invalid add

- processes them through page fault error to the kernel.

- Kernel will check the process
Segmented area & send a signal
called SIGSEGV signal + the
perspective process. On reception
of the signal the process will abort
the operation with segmentation fault.

22

22

24 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a library implemented which has malloc

↳ While this process no debugging we can use Valgrind without debugging symbol info.

↳ Valgrind is not a debugger. It has its own memory allocation & buffer → Printing ASCII values.

↳ Xmalloc is when requesting memory allocation for dynamic memory. Valgrind will allocate for malloc & amount of memory specified. Specified chunk of memory is standard memory actually requesting essence is Configured to request heap memory claim either free upper boundary region or lower boundary region.

↳ Valgrind is a runtime tool to provide protection against configuration essence to suppose worse boundary region violation.

25 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a library which has malloc & free function. Valgrind will print string until you search null character. Valgrind instead of recompiling the code changes can be made from the value of existing command to modifying gdb command to either the GDB command called.

26 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

27 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

28 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

29 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

30 पृष्ठा ३६ ५३ ३०८ ३०९
↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

↳ Valgrind is a runtime tool use to trap down heap memory violations is also a heap memory pushing tool. It is a standard alone debugger.

⇒ System Call es
 → Input / Output

Initializing
atomic

→ atomic-level safety ensures safety of int variable on concurrent access.

first sign of
Excess of
behind not creased
behind too little or no
crease

These add - except will terminate - the current task & success pending task to get execute.

→ If we call shell with tell
about shell awaiting
command → Shell is not available
→ Return 0 → Shell is not available
bcz it's a execute Cmd.
→ When NULL returned →
non zero which means shell
is available.

Thread - Cond - t mycond;

AtomC :-
declaring :-
atomic - t variable name

pthread - cond - wait (pthread -
Cond - t * ptr, pthread - mutex, t
& int);

Tribal Society Variable
tribe वंश जाति लोक
→ used met. safe

Whenever a declared a variable of type atomic in a kernel provides a set of functions to safeguard a declared integer variable.

Wait until the monkey has had
time to eat. Then cut through the condit-

`strategic` is a date type, returning float values.

2029

১৩
৩৫৬ & ৩৫৭

जून 2029
पंडित शाक्त वर्ष १०५४
प्रैकृति
प्रैकृति - Variable Argument
Condit waiting on the
the thread waiting up Signal
Read may will sent a Wake up Signal

Process - Scheduled Parameter of 3 rows

going to particular thread

going policy & storing in process

3rd arg.

In tee process Communication (TCP).

HTTP Client process connect to

HTTP Server process.

173

Concurrent threads

①-Phased - Cond - Broadcast function
if thread wakes up all the
conds which have sleeping on
theseals will be wakable

15

১৫ মে ২০১৪ শার্ট এবং জোড়া

It can do following :-

Default Policy:
→ In this case, prudence is dynamic

- On market, policy persistently
- Fed default risk can be changed by a
dynamic fiscal system based on behavior of the
system based on the needs.

→ Eeo Sch. policy → thread have fires

→ Lowest

- Head with person if in FIFO
Policy will execute list when Computer
to default policy have until
In FIFO it not reschedule until
feminate.

→ RR Sch policy - each & every thread will have - priority & execute in circulate order.
- It sat thread use less service

(node)

→ each machine in the NW is identified by unique IP address (32 bit) & also used for connecting थाके 22 ज्योति communicate. & 16 bit port address is used to identify process in particular node.

→ Linux has provided 5 IPC Comm.

- i) pipe
- ii) FIFO
- iii) message Queue { IPC II std.
- iv) shared memory
- v) semaphores

1) Pipe is a Serial Communication device that permits Unidirectional data transfer - can be used b/w parent & child Process (or) between processes.

Whitelock reached

四

17

29

四

28

- If reader process receives empty header when a pipe is reading then process from pipe as pipe is data from pipe is block until empty heading is written into pipe.
- Same data written into pipe consider getting number on the moment reading a written & start automatically synchronize & pipe audience file after every file receiver gets data.

29

- Types of file :-**
- ① Regular file
 - ② Directory file
 - ③ Pipe file (& fd)
 - ④ Character device file.
 - ⑤ Block device file.
 - ⑥ Socket file.

2

Cscope is a Linux utility used to examine development process to examine this is source code you can examine symbol of a source code. To check what values are initialized & where they reside in your code you can also visit specific function called by what all other fun. You can change structure you can find specific files, you can change by including files you can also used assignment variable option to change the variable value.

```
ctags --list-languages shows all language which supported by ctags.
```

30

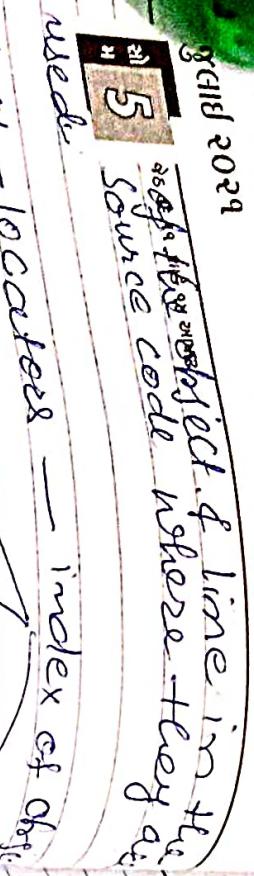
- mkfifo (name of file, mode);**
- wire (open);**
- write (open, mode);**
- read (open);**
- close (open);**
- mkfifo (Name of pipe);**
- Named pipe D**

3

- Ctags is a Linux utility tool which generates a tag file i.e. found in source code & header file. index of object reference to name pathname. the time where is used in the source code. Ctags takes provides a quick references to your source code. It need to find definition of fun & stages uses locators & locates again locate object & pathname.

ग्रेटर २०२१

ग्रेटर २०२१



Stage - locations — index of Obj.
name — pathname — line
tag — Obj / no.
files.

Git Project C3 diff. files.

Git hub Structure.



git clone [repo] [local]
git pull [repo] [branch]

cont:

Working:

Staging:

Repocoding

- modified

- marked as changed

- ready to make

permanent

files.

next level

safety stored

Snapshot files.

→ git configuration files:-

- git config is a configuration variable used to set or get git configuration files.
- --global get in form of `git config --global user.name "John Doe"`
- in local machine.

10

git / GitHub:

git is support disconnected.

Github is different repository.

git is a connected repository system.

Repos Rep.

शार्क शुद्धि शुद्धि शुद्धि

11

git / GitHub:

git is support disconnected.

Github is different repository.

git is a connected repository system.

Repos Rep.

शार्क शुद्धि शुद्धि शुद्धि

11

અને એ શરીર દ્વારા આપેલું હોય
એસ્ટ્રીક્ચર કે એસ્ટ્રીક્ચર

~~the m~~
int mkffofc (const char * pathname
mode_t mode);

Success - 1

- The moment processes executes, executes, creates
mkfifo, a FIFO files get created
& makes entry into Linux kernel
& FIFO file is a Special File.
O.S. Of FIFO file is a Special File.
indicates all attributes. Any process

卷之三

અધ્યાત્મ સુદે ૩ શાકે ૨૨ અધ્યાત્મ

S 3 [15] _____
- ensure that FIFO is open from both ends simultaneously

-If your process is open and

FIFO :
the office sends processes to open

1150 first issue of *Journal of Clinical*

-Affordable cars have multirole

Secularists & progressives & supporters
of socialism & socialism's political party

ପାଇଁ କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା

-file used for Simple client
Server comm. Protocol.

* Message Queue :-
Struct msgqinfo :-

→ defines message for exception
msg of same key.
→ msg send int msgid, const void * ptos size_t nbytes
int flags

The Mönch Preacher in msg
msg to Poincaré pointing to msg
sent to msg whose id Poincaré

- The moment process called msgget system call a new msg queue is created in the kernel space with a unique msg id.

- ~~For newly created msg~~ ^{msg} called ~~msg~~ ^{msg} _{id}

msg id - its allocated in **16** bits

16

1

四

九

卷之三

અધ્યાત્મ ૧૭

msg. Send.

- On Success - msg. is full.

EAGAN - msg. is removed.

EDRM - msg. in interrupt.

ENTER

will receive a msg.

- > msg. for msg. for given thread.
- > from the copy for P + of P + of

- > At msg. (int msgid, int cmd, struct msg_idols) + buf

P
i) IPC-STAT
ii) IPC-SET
iii) IPC-RMID

shared Memory (S.M.)
P₁
P₂
P₃
P₄

U.S.
P
અધ્યાત્મ ૧૨ શાકે ૩૦ અધ્યાત્મ

અધ્યાત્મ ૧૬

20
અધ્યાત્મ ૧૧ શાકે ૨૫ અધ્યાત્મ

અધ્યાત્મ ૧૦ શાકે ૨૮ અધ્યાત્મ

100
memory

- showed memory :- one of the ipc techniques
- sh. m. allows two or more process to access the given region of s.m. & s.m. is faster than region of s.m. because bcz didn't need to jump from one memory location to another while data spread over open between reader & writer process or client & server.
- s.m. can be used betn server & client only & this is synchronizing access of the given regions with semaphores. each & every s.m. will have a structure called shared-mem-struct.
- #include <sys/types.h>
<sys/stat.h>
<ipc.h>
<shmem.h>
- shmat (int shmid, void *addr,
int flags);
- When process execute shmat - system call will map the given region of the given process at a space (addr). provide a flag (shm-RDONLY).

અધ્યાત્મ ૧૮

અધ્યાત્મ ૧૯

અધ્યાત્મ ૨૦

અધ્યાત્મ ૨૧

અધ્યાત્મ ૨૨

અધ્યાત્મ ૨૩

અધ્યાત્મ ૨૪

26

Shmidt C Veld * older);
Shmidt C Veld \rightarrow sections by Shmidt

ipcom -m id
fpcrm -m key

\rightarrow Shmuel Cshmid, int mech. Struct
electrical eng prof).

ଅପ୍ରାଚ୍ୟତ ମନ୍ଦ ଶାର୍କ ପ ଖାଲୀରେ
- - - - -

Linux kernel architecture

U.S. process. file 110
Memory

192

1. *Leucania* *luteola* (Hufnagel) (Fig. 1)

१८ अप्रैल १९५४ शुक्रवार दिन चूल

5

```

graph TD
    Process[Process] --> MS[M&S]
    Process --> Management[Management]
    MS --> MarketResearch[Market Research]
    MS --> CompetitorAnalysis[Competitor Analysis]
    Management --> StrategicPlanning[Strategic Planning]
    Management --> OperationalEfficiency[Operational Efficiency]
  
```

→ **Memory** **Storage**

so heavy as

device officer do
by collection

Computer Architecture

System calls are kernel's space services not user space services spread to invoke K.S. They are spread every system call has to pass through System call interface ~~upper~~ appropriate K.S. Service

Advantages from Kernel's point of view
it gives a piece of software that can manage hardware & brings the functionality of device driver

In Linux OS device ~~management~~
is done in the K.S. bcz only
the K.S. driver has privileges to talk
to the hw.

30
50

29
50

→ In Linux OS device ~~management~~
besides in the K.S bcz only
the K.S driver has privileges
to the HW.

→ Linux has two interfaces Q. interface
bct appn & device driver i.e., OS Specific
& Q. device of HW i.e., HW implement
at.

→ Semaphores are one of the IPC Com.

area of Sonoran ^{and} ~~Arizona~~ desert.

Issue brief: Large Open Software Process needs to work on lots of

Resources needed may be pasted from
anyone of several sources.

Demeter, Semele, Europa, Persephone, etc., still are designate for
work as a day of Semperphore.

#~~5895~~ / types. h > Sys.h > Syspc.h > Sys/cem.h

卷之三

ପାତ୍ରମାନ କିମ୍ବା କିମ୍ବା କିମ୍ବା

(acmealmy)
spreading
finestring

Programming
by Example

Agaricaceae

٦٥

at debugging. :- process tracing is called debugging.

compiled at Compita, Firmino,
Conapile, Cava, Resage, &c. &c. from
varia & infirmable evidence.

Failure: When a system specified fails to perform specified functions leads to failure.

↳ **Java** library in compatibility.
↳ **Type check**.

- * fault :- It is a condition where system software fails to perform an action.
- * static Code analysis.

deviations from prescribed coding standards.

It is the process of handling program memory before program execute.

Linking program prescribed coding
decades.

मात्रा संख्या १३ शास्त्रीय वर्ष शास्त्रीय वर्ष

gcc -E one.c → one.i → preprocessor
gcc -S one.i → one.s → assembler
gcc -c one.s → one.o → compiler.

Linker Responsibilities:
→ Linker job is to provide runtime linker code.

Runtime code

- It is done best on set by saying some coding standard. This kind of analysis will help to prevent the scope holes. The neatness in writing code that might be harmful.

શાખા સૂચના પત્ર નાંદું ૨૦ અધ્યાત્મ

Standardizing Program Prescribed coding

deciding standards,

gcc -E One. C → One. i → preprocessor
 gcc -S One. i → One. S → assembly
 gcc -C One. S → One. o → compiler.

Linker Responsibilities:

- linker job is to provide runtime code

a.out	Runtime code
executable file.	<ul style="list-style-type: none"> - init - start - final
functionality	sections
library	symbolic references

શાખા સૂચના પત્ર નાંદું ૩૦ અધ્યાત્મ

21

- Analysis is done on ~~Structurally~~
- piece of Software - ~~that~~
it is like static code analysis

प्रोग्राम प्रोसेसिड कोडिंग
स्टैंडर्ड्स

भाषा ३० १३ शाके २० भाषा

gcc -E one.c → one.i → ~~preprocessor~~ १००
 gcc -S one.i → one.s → ~~assembler~~ १०२
 gcc -C one.s → one.o → compiler.

Linker Responsibilities:

- Linker job → to provide runtime code

```

graph TD
    A[Executable file] --> B[functions]
    A --> C[libraries]
    B --> D["Runtime code"]
    C --> D
    
```

भाषा ३० १३ शाके ३० भाषा

22

21

- Splint is a static code analyzer. It is used to identify the problematic areas of suspicious code and suggest corrections.

અધ્યાત્મ પ્રોગ્રામ પ્રોક્રેસીબલ કોડનું પ્રદાન કરું

```

graph TD
    LJ[Linker job] --> C[Code]
    LJ --> R[Responsibilities]
    C --> OC[Object code]
    C --> E[Executable file]
    OC --> F[Functionality]
    OC --> L[Library]
    F --> RC[Runtime code]
    L --> RC
    RC --> I1[init]
    RC --> S1[start]
    RC --> F1[final]
    L --> RC
    L --> RC
    RC --> I2[init]
    RC --> S2[start]
    RC --> F2[final]
  
```

Linker Responsibilities:

- Linker job is to provide Runtime Code

Object code

Executable file

Functionality

Library

Runtime code

- init
- start
- final

Runtime code

- init
- start
- final

Start :- main() :-

Init :- initializes key resources for execution.

Final :-

22

2

અધ્યાત્મ પ્રોગ્રામ પ્રોક્રેસીબલ કોડનું પ્રદાન કરું

卷之三

શ્રાવણ વદ ચ શાકે ચ ભાષ્યમે

Code
ladder address
in train

(只限于 1995 年以后)

ଶାବ୍ଦିକ ପତ୍ର ୧୧ ଶାବ୍ଦିକ ୧୧ ଭାବୁମ୍ବା

୭

- Kernel long division :- It starts from Boot strap Code. It starts with the CPU reset

10

init exec (login)

1

execute & share. It is
real made to predict & order
& the process class process on
initialization.

login

2

~~RAM~~ ~~Kernel image.~~

Posky shell.

શ્રાવણ વદ ૧૭ આદે ૧૨ અન્યાં

2

ମୁହଁରା ପାତାର କିମ୍ବା କିମ୍ବା କିମ୍ବା

Posky shell

शालवा वड ११ शाहे १२ लाद्यांद

3

• • • • • • • • • • • •

Rocky shell.

શાખા વદ ૧૭ શાકી ૧૨ લાદ્યા

3

શાખા

→ The tree has 3 boart pions. It has
 $2^3 = 8$ boarting options.
If 5 boart pions then $2^5 = 32$ boarting options.

- Kernel is a piece of code loaded into RAM execute until it starts own initiative.

Basic shell

આપણ વદ ૧૧ શાહે ૧૨ ભાદ્યાદ

→ There are 3 boot pins it has
 $2^3 = 8$ booting options
→ If 5 boot pins then $2^5 = 32$ boot options
→ dynamic → of Analysis of Source Code
Gcc → Gcc Coverage tool

3 શૃંગ

Die Verhältnisse sind
sehr schwierig.

Booting shell.

શાખા પદ ૧૧ શાહે ૧૨ ભાગ્ય

3 शु.

→ There are 3 boot pins. It has
 $2^3 = 8$ booting options.
If 5 boot pins then $2^5 = 32$ options.

→ Dynamic tool Analysis of Source Code
GCC → Gcc Coverage Tool.
Open Source tool.
gdb or DWARF along with GC.
→ Given in next few number of source

શાખા પદ ૧૧ શાકે ૧૨ ભાગ્ય

Posh shell:

→ These are 3 boot pins it has
 $2^3 = 8$ booting options $2^5 = 32$ boot pins
If 5 boot pins then there are 32 options

→ Dynamic tool Analysis of Source Code
Coverage tool
Open Source tool:
GCC v DWARF along with GDB.

- GDB is used for analysis of Source Code if it will check for undefined part of Source Code
Identify non-executable code

शास्त्राचार्य १० श्रीकृष्णानन्दस्मृति अथवा विश्वामित्र एवं क्षेत्रेन
३५
१) Setup memory DOS.
२) - - - interrupt DOS.
३) - - - device DS.
४) - - - process table.
५) - - - memory.

Posh shell

શાખા પદ ૧૧ શાકે ૧૨ ભાગ્ય

3

→ There are 3 boot pins it has
 $2^3 = 8$ boot pin → Then $2^5 = 32$ boot options
if 5 boot pins → Then
→ Dynamic tool Analysis of Source Code
GCC → gcov → coverage tool
open source tool.
gcov is used for analysis of source code & its will check for run time part of source code
Identify non-executable code
Gcov also use as profiling tool.
- Can be used for monitoring & managing system
- Allows you to modify & examine
source code.

4

v) - Initialize CPU Scheduler
vi) Initialize Kernel Thread

Posh shell	
સાધણ પદ ૧૧ શાકે ૧૨ ભાષય	3
સાધણ પદ ૧૩ શાકે ૧૪ ભાષય	4
5	શાધણ
→ Compiling Source code with Heron	→ These are 3 boot PIns it has $\frac{2^3}{2^3} = 8$. booting options if 5 boot pins → Then $2^5 = 32$ boot options
→ Dynamic tool Analysis of Source Code GCC → gcov . coverage tool open Source tool	→ gcov is used for analysis of Source Code & its will check for runfa part of Source Code Identify non-executable Code Can also use as profiling tool. - Can increase & many other - allows you to modify & analyze source code.

સપ્ટેમ્બર ૨૦૨૧

સો
મ

6

શાકે ૧૫ ભાડ્રપુદ

2nd flag do some analysis of source code & generates a hidden code in binary & also records how many times code is executed.

+ 5th flag generating a Profiling file with .gda & .gcov.

મં
ગ
સ

7

ભાડ્રવા સુદ ૧ શાકે ૧૬ ભાડ્રપુદ

બુ
ધ

8

ભાડ્રવા સુદ ૨ શાકે ૧૭ ભાડ્રપુદ

Signal handlers:-

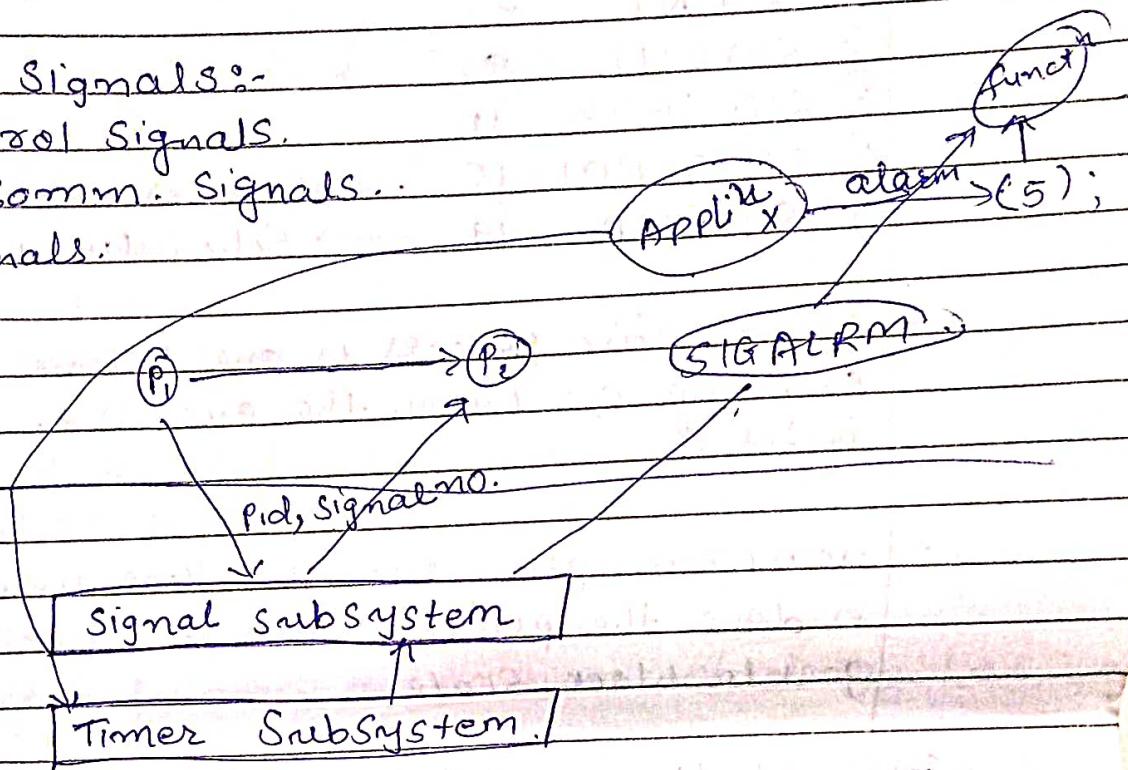
```
Void (* signal (signo, void (* fun) (int))) & (int);
```

→ Types of Signals:-

- i) Job Control Signals.
- ii) Process Comm. Signals..
- iii) I/O Signals.

U.S.

K.S.



→ In the kernel there is a S.S.S. which is responsible

* for delivery of Signal in each process of U.S.

Case: 1 → if Appli^m 1 Wants to send Signal to Appli^m 2 it has to register S.S.S. & convey the Appli^m id & Signal no. to which Appli^m 1 Want to send a Signal.

→ S.S.S. on behalf of Appli^m 1. deliver the Signal to Appli^m 2.

Case: 2 → Appli^m x Wants to perform a periodic task fun

then registers with a timer SubSystem for delivery of Signal at a particular time out.

→ T.S.S. Collects the time slices. C Timer S.S. invokes the S.S.S. after some time out & S.S.S. will then deliver a Signal called SIGALRM to Appli^m x.

i) In S -

- ① SIGINT 2.
- ② SIGKILL 9.
- ③ SIGSEGV 11.
- ④ SIGTERM 15 → Process terminate.
- ⑤ SIGSTOP 19 → Schedule to process preemption.

- Preemptive process is one kernel forcibly shifting CPU from the process & with out interrupt.

- non preemptive process : Here until the time slice is done the process will not leave CPU or process goes to ready state.

ii) SIGKILL & SIGSTOP Signal cannot be handled or caught.

- i) Process :
- ① SIGUSR1 → Process +→ Process
- ② SIGUSR2 → Process
- ③ SIGALRM

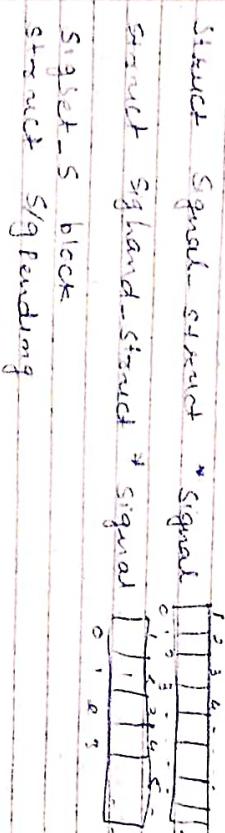
iii) Task :

This delivered by IP subsystem to the process listen a particular fd on sockets are ready to perform input output.

→ signal(SIGINT, SIG_IGN);
 ^simply ignore.

Execute, process will not resume.

K.S → PCB → Struct having task_struct.



Struct Signal_struct * Signal

Struct Signal_struct * Signal

→ execute " default handler :

In signals struct is a fun " doSendSignal() " this fun is executed before delivery of a signal.

→ steps followed by send - Signal.
In we get reference of PCB to which sig has to be delivered.

Now we will get point to the fun received by PCB i.e., pointing to a vector of 64 elements.

v) send - signal fun will manipulate the Signal_structs . for ex: SIGINT manipulating 2nd element of vector. 0 is disabled & 1 is enable after setting send - signal will seize CPU except

the CPU step current task & CPU executes

exceptn handler fun.

vi) Exceptn handler fun now access the 2nd struct of PCB i.e., Signal_struct again. Signal_struct pointer pointing to vector of intag Signal_handler locate for the Signal_struct wire exceptn handler locate for the Signal_struct & map to respective fun for enable element & maps to respective fun in Signal_handler fun.

vii) Now exceptn handler executes the default fun in the interrupt handler until the handler

if when default "SIG_BLOCK" is executed in interrupt context then Context switching is not possible.

→ Process blocking Signals :-

A process needs to block signals.

for ex - a process dealing with Critical Section & Critical Section overlapping same database during this process does not want to preempt then process can block particular signals.

→ option X blocked signal X, applying want

to fire the same signal
because once same the signal X move to the PCB of Application & then places the signal X into sigpending Structure;

signal X will be released only when the application X will unblock the signal.

→ sigemptyset (sigset = *set);

- sigemptyset again empty the structure & initialize it.

→ sigfillset (sigset = *set);

- initializing with all the element in struct

→ sigaddset (sigset *set, signo),
adding signal one by one.

→ sigprocmask (flage, sigset-t *set, sigset
current *set, *oset);

old

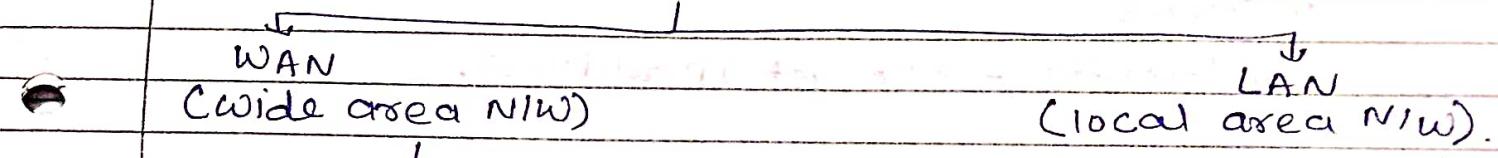
- sigprocmask will block & unblock the sigset from the block & sigunblock flags.
- based on sigblock & sigunblock flags.
- at this point of code kernel will send signal & placing it in pending struct & never send deliver later when signal is unblock.

→ sigismember (sigset = set, signo);
- it is checking the signal is member of sigset or set argument or not.

Network Programming

⇒ N/W APPⁿ programming is all about writing APPⁿ programs using N/W APIs & N/W protocol stacks. This program communicate with program either on the same machine or on different machine at remote locatⁿ.

Computer N/W.



CSN (Cit.s.N.).

(dedicated

Physical link

Establish betⁿ

through P.E. nodes,

- ex:- telephone

- Path is connectⁿ of

Sequence of physical

links betⁿ nodes.

(cable).

- Sender wants to

transmit packet so,

packet goes through

this link rapidly as.

PSN
(Packet Switch N/W)

Protocol.

- that devide data

into smaller packet

c/a N/W Packets.

- This are transmit

out n/w.

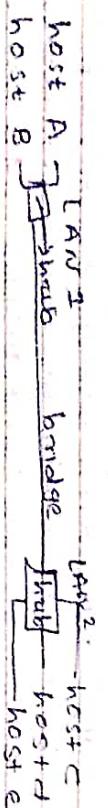
- digital formate. (0&1).

- ex:- wifi

* N/W devices:-

i) Router :- switches :- N/Wing devices they are used to connecting one or more devices to other computer or devices or to connect other N/W.

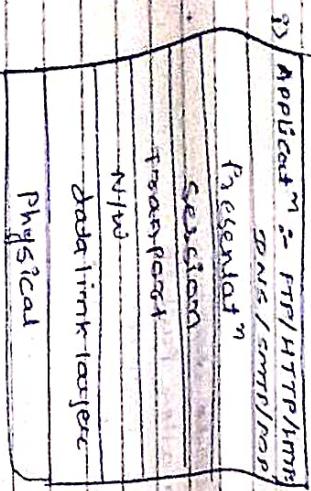
v) hub is used to connecting devices in a LAN.



vi) Bridge: It is a networking device, connects two LANs.

vii) server: - it configures to get IP address.

sender →



viii) It decides in which format packet will sent to receiver.

- Provides encryption & decryption, for security
- Compresses size of packet so, packet transmit faster.

ix) It establishes connection i.e., it is creating session between Sender & receiver. until receives receive packet.
- maintain synchronization & terminate fun.

- maintain synchronize & terminate fun.

- While transmitting, packet from session layer bcz of some event goes down layers step by step transmission continues until reaches to system. Organize continuous maintaining form part of interconnection.

x) It divides data into smaller segment (size 1500).

- i) M.W. → decides path & enrich data transmitted
- converts data packets to network packets
- It has IP add. of src & dest.
- All routing process starts from M.W. layer
- All routing takes place then send to M.W.

xii) Create frames

- checking for errors, whether data is error free or not. If errors removes the errors & transfers error free packet.
- maintains data rate speed bet' sender & receiver like sender 100 mbps & receiver 100 mbps then traffic n at receiver side receiver may loose packet.
- maintain common database bet' sender & receiver
- physical addressing job

- xv) convert packet in os & its digital formate
- we have networking devices will convert data into electrical signal or radio wave.
- Session bet' Sender & receiver. until receives receive packet.
- "is coming interface configured"). Will display active NYU interfaces of your system.

- MAX → max. Transmit Unit i.e. max size of packet in a transmission.
- Frequency → max. no. of packet in transmission
 - ↳ it is very dedicated to Linux Kernel that deals with NW devices.

loopback add - each device has loopback add. 127.0.0.1.
 It's also via local host when we transmission data using loopback add. Data never reaches the NW card interface itself.
 (loopback add is used for passing traffic)
 It will help the device to send & receive its own data.

- ifconfig is used to view & change NW interface configuration of your system.

choice of APIs

↳
 API key secret
 API

• A Real time Appln:

- So called System Call:
- CSC supports many NW Comm. We can say that CSC designed to support NW Comm. Protocols.
- But at same reason CSC packets are generic in nature.

- As we are using generic S.C. all the S.C. will use same socket structure as argument [struct sockaddr].
- S.S.C. takes size of parameters. This will identify size of socket structure.



PAGE NO. _____ DATE _____

Server

```

    Socket();
    bind(); → adding Src
    socket(); → IP address &
    port addrs
    half binded.
    listen();
    listening for once connect
    Client conn. Request
    Client, Server set to
    while loop. Servers in
    blocking opn
    accept();
    has to accept client
    accept();
    
```

Client

```

    bind (int socket id, struct sockaddr *serv,
    int addrlen);
    connect();
    → int listen (int socket id, int backlog);
    Server
    listening for once connect
    establish comm
    starts betw client
    & server
    
```

To-fair sockets are internally fair.

→ bind (int socket id, struct sockaddr *serv, int addrlen);

→ int listen (int socket id, int backlog);

→ accept (int socket id, struct sockaddr *cli, int *addrlen);

Client side

→ int socket (int family, int type, int protocol);

→ connect (int socket, struct sockaddr *serv, int addrlen);

Sender → NIO byte order.

Host → APJ's Rcv.

Host → APJ's Rcv.

short Sint_Family;

→ decide my protocol type.

USHort Sint_Port;

→ 16 bit port no. (my byte order).

Struct in_addr Sint_addr;

→ 32 bit IP address. conv. byte order.

→ few → small C fd).

→ Socket Client family, int type, int protocol);

AF_INET (Type of Protocol) has 0 address format comm. value in IP protocol.

Internet Protocol (IP) → adder (header).

Raw_Socket. (Socket Stream).

SOCK_DGRAM

NIO byte order.

→ which are using different types of computers

& trailers. few CPU follows big endian format

& few follows little endian to store integer

byte in memory.

TCP/IP specifies all the protocol headers

as a NIO packet should follow NIO byte

order. C Big endian format)

every machine in the NIO should be aware

of this & should convert port order to

TCP
UDP

- again at the receiver side while receiving incoming packet should

- correct N/w byte order to host order

- source code size API's to convert host to n/w order & vice versa to match respective CPU requirements.

- This API's works for both little endian

& Big endian

- host to n/w long sum taking 32 bit host

as address IP add. convert into 32 bit

among them. Coming host long;

- In case of host short;

host → host port. 16 bit to host.

→ connect establish → No connect establish
 → connect oriented → connect less

→ slow → fast

→ reliable b/cz acknowledge → not reliable b/cz no

extensive transmission in case of data loss

→ extensive error checking → basic error checking

→ streaming byte → client before transmission

→ header size 20-60 bytes → 8 bytes

→ No broadcast support → broadcast support

→ supports data sequencing → no data sequence

N/W pkt arrived in order.

→ HTTP/HTTPS/POP3/FTP

→ TCP can guarantee seq

data delivery → no guarantee.

→ for connect establish → no 3-way

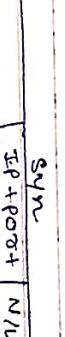
3-way handshaking → no 3-way

for connect close → no 4-way

handshaking → 4-way handshaking.

→ 3-way handshaking:-

server → client



→ server open port & accept client initiation request

(Syn Disc. Process PKT)

Response → connection confirmation → ACK

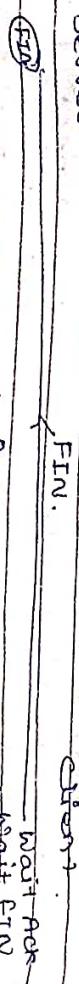
→ server ACK → client

→ ACK.

→ client

4 way handshaking

Server



Types of servers.

Interactive Server :- It shows one client connected at a time and client in the queue waits till client 1 has terminated only then server serves.

Session Server :- After accept system call a new client socket created of same type now code extends to while loop & server receives the client data & processes the data. New program come out of inner while loop only when client closing the connection (n=0).

Once client closes server again executing accept system call & awaiting from next client.
It is a simple interface
If each client transaction duration is less then it's good.

- Implementation took in servers to handle a client request.

→ ssize_t recvfrom (int sockfd, char *buf,

int length, int flags,

struct sockaddr *addr, msg *msg, /* corner */

int addrlen);

↳ This is source from where we have to take data.

→ TCP

→ connect oriented

→ stream data transfer

→ FIFO

→ connect n less

→ reliable

→ non reliable

→ unreliable

→ ssize_t sendto (int sockfd, char *buf,

int length, int flags, struct sockaddr *

addr, int addrlen);

↳ This is destination address where we have to send data.

→ UDP

→ connect less

→ non reliable

→ unreliable

\$ netstat

-tulpm
tcp & udp listen pid of process

open source tool classifies tasks across how your computer communicating with other computers or other networking devices.

& also provides detail information of individual NW comm. also provides info. used for troubleshooting or diagnostic purpose.

- A file called services in etc folder displays all the services used by client APPL in your computers.

- A fun called getservert will read the next line of services folders in etc folder & return a pointer to next entry in services file.

& RAW-SOCKET

PACKET SOCKET = SOCKET (PF_PACKET, type, (ETH_P_ALL));

SOCK_RAW

SOCK_DGRAM

→ structure belongs to data link layer

structure have family, protocol, physical packet type, header file, physical Adr. & Adr.length.

- struct sockaddr_ll.

→ see family; → PF_PACKET;

see protocol; → ETH_P_ALL.

see ptchtype;

see hatype;

see address → physical link layer.

SOCKET_RAW::

SOCKET_AF_INET, SOCK_RAW, 0);

Raw socket sends packet to specific users. user will get untempered final NW packet.

This packet is bypassing all TCP/IP processing for optimizing & customizing WAN.

If you want to improve performance & security on critical traffic work load on WAN Raw sockets can be used.



SLI_addrlen;
}

→ PF_PACKET family used to create packet socket.

- data transmission is from physical link layer i.e., below internet protocol layer.

→ Type can be SOCK_RAW or SOCK_DGRAM:-

- When use SOCK_RAW packet including link layer header is pass to the specific user.

- When use SOCK_DGRAM, packet without link layer header is pass to the specific user.

- Protocol value define by IEEE member.

- Std. ethernet protocol. (ETH_P_ALL). 800.

→ int setsockopt (int sockfd, int level, ^{SOL_SOCKET}
 int option_name, ^{SO_REUSEPORTS}
 char * option_value, ^{0 default}
 size_t size_of_option_value);

- Setsockopt is used by Applⁿ program to change socket behaviour. we can set any option of socket.

- int level :- request from socket level.
 ---> ---+ transport level.
 ---> ---+ N/W layer.

→ viele verschiedene Werte und
verschiedene Verhältnisse zwischen ihnen, es kann
sehr lange dauern bis P/P konstant oder fast konstant
geworden ist.

To overcome this defect, we have proposed the following
to avoid from multiple type mistakes.
we can take one of the following methods
and in each,
select a system that makes the program simple to
read from multiple, so, when it is done, we
can it to operate.

internal data transmission → shared written principles

red portion - TCP Connection

- ssh is used to connect two machines over the n/w & also to control the remote machine
- 2 scp is transferring file.

→ if scp cmd n/w address : /home/username
→ after establishing the connection b/w client + ssh & server ssh, ssh protocol is providing strong encryption & algorithms for ensuring the privacy

- 2 data security b/w client & server machine.
- IPsec: - The only one system can access user space that can access the data port & cmd. ports of particular device.

wireshark :- It is a open source software tool that captures incoming & outgoing n/w data packets in real time & captured information is used for n/w troubleshooting & diagnosing & packet analysis, SW development & comm. development protocol.

- with wireshark we can see the things around n/w. & get detailed packet moment information.
- wireshark supports 100 of packets of different protocol. only logic applied is capture packet online & examine them offline.
- It can read & write different captured file formats.

view:-

- main toolbar
- filter bar is used to filters required packet
- status bars show summary (selected & last packet).

5/4/2022

STM32L475VG :-

- Core :- Arm Cortex-M4
- ⇒ Series :- STM32L4
- ⇒ Line :- STM32L4X5
- ⇒ Package :- LQFP100
- ⇒ Other :- Price = 5.48

IO = 82

Eeprom = 0 Bytes

Flash = 1024kBytes

Ram = 128 kBytes

Freq. = 80 MHz

- ⇒ Peripheral :- ADC 12-bit, DAC 12-bit, I₂C, LPUART, CAN, DFSDM, IRTIM, OPAMP, COMP, FMC, LPTIM, QVADSPI, RNG, RTC, SAI, SDMMC, SPI, SWPPI, Timer - 16-bit, Timer 32-bit, Touch Sensing, UART, USART, USB OTG FS