Linux Kernel:

Answer-01

Booting Process:

It consists of

BIOS

MBR

Grub

Kernel

Init

Bios: Bios stands for Basic input/output system.It will perform system integrity checks and it will searches and loads and executes the bootloader program.

MBR: It stands for Master boot Recorder.It is located in the first sector of bootable disk.It will loads and executes the GRUB Bootloader

Grub:Grub stands for Grand Unified Bootloader.Grub displays splash screen,wait for few seconds,If we dont enter anything it loads the default kernel image.

Kernel: Kernel contains neccessary drivers compiled inside, which helps it to access the harddrive partitions and other hardware

Init:Init is the first process wich will be executed.

Answer-02

stackoverflow:  
 The stack area grew the opposite direction of heap area, when the stack pointer met the heap pointer, free memory was exhausted.When the size of the stack was full and still the memory has to be allocated.At that time stack overflow will occur. For example Calling function internally other function every time stack frame will be created.but not free that call internally.So if this happen continously than stack overflow will occur.

Answer-03

If there any errors in the allocation of the memory, i.e if the memory is not utilized properly than that is called memoryleak.Valgrind tool will be used to find the memory leaks in the code.

Answer-04

Mutex:

Mutex provides one person to access a single resource at a time, others must wait in a queue. Once this person is done, the guy next in the queue acquire the resource.So access is serial, one guy after other.

Semaphore:

Semaphore are useful if multiple instances (N) of a resource is to be shared among a set of users. As soon as all N resources are acquired, any new requester has to wait. Since there is no single lock to hold, there is as such n ownership of a semaphore.

Answer-05

A core dump is a file containing a process's address space (memory) when the process terminates unexpectedly. Core dumps may be produced on-demand (such as by a debugger), or automatically upon termination. Core dumps are triggered by the kernel in response to program crashes, and may be passed to a helper program (such as systemd-coredump) for further processing. A core dump file will be passed on to developers upon request.It will give the information of the program state at the time of crash.core dump file will be useful if the fault is hard to reliably reproduce.

Answer-06

1.A page fault occurs when the requested memory address (from the virtualaddress space) does not map to something that is in RAM.

2.A page need to be sent from RAM to swap, so that the requested new page can

be brought from swap to RAM.

3.When the referenced page number is not in Main Memory, Page fault occurs.

It decreases CPU utilization

Answer-07

Dead lock:

Suppose cosider there are three resources like resource1,resource2,resource3.let us consider process1 has acquired the resource 1 and process2 has acquired the resource2 and resource3.So now if the process1 want to complete it’s execution it must acquire resource3,But the resource3 was acquired by process2. And if the Process2 want to comlete it’s execution it must acquire resource1 but it was acquired by process1.So both the process will go to waiting state.This type of condition is called deadlock.So inorder to overcome this type of problems Locking mechanism is used.

Answer-08

Static library:

In static library all the dependency files are loaded during the compile time.Due to this The size of the file is large

1. Static Memory Allocation:

The static memory allocation is normally used you know how much memory space you’ll need.

define BUF\_LEN

2048

char buf[BUF\_LEN];

However, the kernel stack size is fixed and limited (the limit is architecture dependent, but normally

it’s only tens of kilobytes). Therefore people seldom request big chunk of memory in the stack. The

better way is to allocate the memory dynamically from heap.

Dynamic library:

In Dynamic library all the dependencies are not loaded.Only references will be loaded.So the speed of the execution will be slow when compared to static.The size of the file is small.

Dynamic Memory Alloation:

There’re two functions available to allocate memory from heap in Linux kernel process,

vmalloc

The vmalloc function is defined in /lib/modules/$(uname -r)/build/include/linux/vmalloc.h

as below,

void \*vmalloc(unsigned long size);

Answer-09

1. An interrupt is an unexpected hardware initiated subroutine call or jump that temporarily

suspends the running of the current program.

2. Whenever device driver raise interrupt, isr should process fast and respond to interrupt

immediately.

3. After creating data structure for device that called top half and later process other works related to

driver in bottom half.

4. Lengthy tasks inside interrupt handler degrade system responsiveness.

a. Top Half: Perform time critical tasks such as acknowledging receipt of interrupt, reseting

hardware etc. It is the routine that actually respond to the interrupt –the one you register with

request\_irq().

b. Bpttom Half: Perform any interrupt related work not performed by interrupt handler. he bottom

half is a routine that is scheduled by the top half to be executed later, at a safer time.A bottom half is

a low-priority function, usually related to interrupt handling, that is waiting for the kernel to find a

convenient moment to run it.

5. The big difference between the top-half handler and the bottom half is that all interrupts are

enabled during execution of the bottom half—that’s why it runs at a safer time.

6. In the typical scenario, the top half saves device data to a device-specific buffer, schedules its

bottom half, and exits: this operation is very fast.

7. The bottom half then performs whatever other work is required, such as awakening processes,

starting up another I/O operation, and so on.

8. This setup permits the top half to service a new interrupt while the bottom half is still working.

Answer-10

While the program is executing when the kernel doesn’t know to wich process to be executed next.Than at that time kernel panic will occur.If the kernel panic will occur than the execution of the program will stop unconditionally.

Python:

Answer-02

#!/usr/bin/python

def sum\_xor(li1):

res=0

length=len(li1)

for index in range(length):

for index1 in range(index+1,length):

res+=(li1[index]^li1[index1])

return res

li1=[7,3,5]

li2=[5,9,7,6]

res=sum\_xor(li1)

print "The sum of xor of all pairs of numbers in array is:",res

res=sum\_xor(li2)

print "The sum of xor of all pairs of numbers in array is:",res

Output:

