

AOS - Assignment.

Page No. _____
Date _____

① Define the

Justify the following process can exercise crude control of their scheduling priority by using nice () system call function.

→ The kernel implements a fair-share scheduling algorithm that gives processes a share of CPU time based on priorities assigned to them depending on the nature of the task. Higher priority process get scheduled more often and receive more CPU time.

process priority is a function of this nice value.

process priority = recent CPU usage / a constant + base priority + nice value.

This algorithm gives user group A twice the slot for group B, three times that of C, and four times that of D, where user processes are grouped by priority.

② Justify the following process 0 and process 1 exists through the lifetime of a system.

→ The PID 0 is reserved for the swapper process and 1 for the init process. The startup function for the kernel establishes memory management, detects the type of CPU and any additional functionality such as floating point capabilities, and then switches to non-architecture specific Linux.

Kernel functionality via ~~add~~ call to start_kernel(). Init is the father of all processes. Its primary role is to create process from a script stored in the file.

(5)

- ③ Justify the following At the kernel level, support for protected process is two-fold.

→ At the kernel level support for protected process is two-fold. first the bulk of process creation occurs in kernel mode to avoid injection attacks. second protected process have special bit set in their EPROCESS structure that modifies the behaviour of security related routines in the process manager to deny certain access rights that would normally be granted to administrators.

- ④ In Linux files are usually accessed via file names.

→ In Linux files are usually accessed via filenames, they are not directly associated with such names. Instead a file referenced by an inode which is assigned unique numerical value. The value is called inode number or ino.

- ⑤ Explain programmatical behaviour of main ()
- ```
int status;
if (fork () != 0)
 exec (" /bin/date", "date", 0);
 wait (& status);
```
- The fork system call in Unix creates a new process. The new process inherits various properties from its parent. That is Environment Variable, file descriptors etc.
- After a successful fork call, two copies of original code will be running in the original process that is the pattern.
- when a type "date" on the unix command line, the command line interpreter i.e. "shell" forks so that momentarily 2 shells are running
- ⑥ write a program to open file in write append mode. Suppose the size of the file system in byte. At the (n+100) byte the same file, write the string "UNIX".

→ #include <stdio.h>  
Void main().

```
 int fd;
 char *buf = "VNJX";
 fd = open ("file.C", 'a');
 if (fd == -1)
 perror ("error");
 lseek (fd, 0, SEEK_SET);
 lseek (fd, 100, SEEK_CUR);
 write (fd, buf, 5);
 close (fd);
}
```