

SP_S17_Morning_Quiz01_Solution

Questions 01:

a)

For this type of race condition kernel provides a simple solution. auto-append mode. When the O_APPEND bit is set for a file descriptor each call to write implicitly includes an lseek to the end of the file. Moreover, the kernel combines lseek and write into an atomic operation.

```
int s = fcntl(fd, F_GETFL); //get settings
s |= O_APPEND; //modify settings
fcntl(fd, F_SETFL, s); //set them back
write(fd, &rec, size)
```

b)

c)

From man readdir:

If the end of the directory stream is reached, NULL is returned and errno is not changed. If an error occurs, NULL is returned and errno is set appropriately.

```
DIR * dir_ptr = opendir("/tmp");

struct dirent * dir_entry = NULL;

errno = 0,
while (dir_entry = readdir(dir_ptr))
{
    printf("%s\n", dir_entry->d_name);
}

if (errno && dir_entry == NULL)
{
    perror("readdir() failed");
}
else
{
    printf("No more entries.\n");
}
```

d)

mknod <name> <type of device c:- character b:- block device etc> <major> <minor>

```
mknod chfile c 89 1
```

e)

```
int currpos;
currpos = lseek(fd, 0, SEEK_CUR);
```

```
f)
umask = 0123
~umask = 654
      633  110 011 011
&    654  110 101 100
-----
      610  110 001 000
and that is
      rw- --x ---
```

g)

```
fd = open("/tmp/file1.txt", O_WRONLY | O_CREAT | O_TRUNC, 0666);
```

Question No 2

```
void show_file_type(char* fname){

    int i;
    struct stat buf;
    char * ptr;

    if (lstat(fname, &buf)<0){
        perror("Error in lstat");
        exit(1);
    }
    printf("%s",fname);

    if (S_ISREG(buf.st_mode)) ptr = " is a regular file";
    else if (S_ISDIR(buf.st_mode)) ptr = " is a directory";
    else if (S_ISCHR(buf.st_mode)) ptr = " is a character special file";
    else if (S_ISBLK(buf.st_mode)) ptr = " is a block special file";
    else if (S_ISFIFO(buf.st_mode)) ptr = " is a FIFO file";
    else if (S_ISLNK(buf.st_mode)) ptr = " is a soft link";
    else if (S_ISSOCK(buf.st_mode)) ptr = " is a socket";
    else ptr = " have unknown mode";
    printf("%s\n",ptr);

}
```