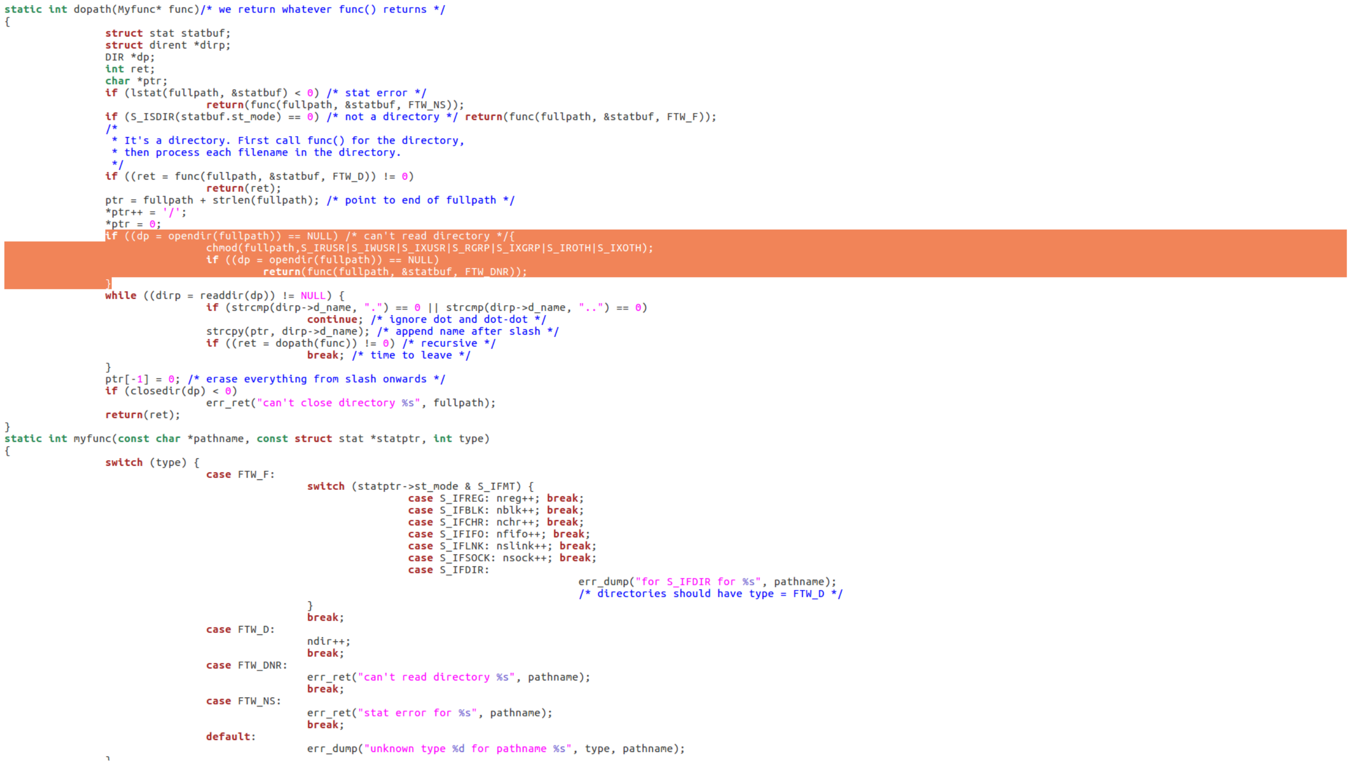
系統程式設計
Lab3 File Attributes

學號：

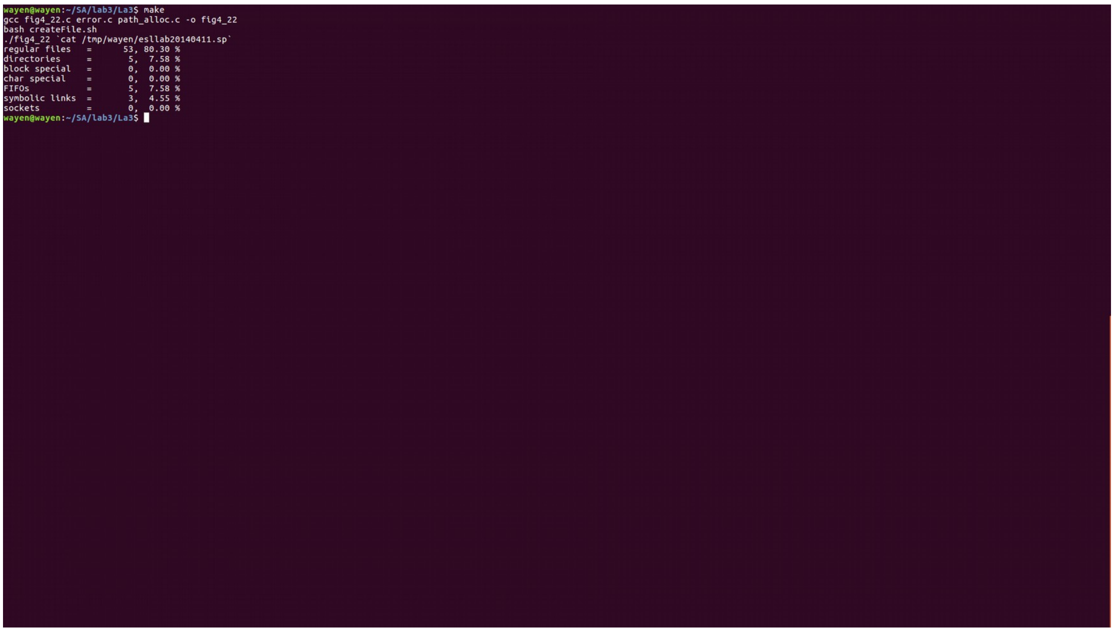
姓名：

I. How do you modify fig4_22.c?



```
static int dopath(Myfunc* func)/* we return whatever func() returns */
{
                struct stat statbuf;
                struct dirent *dirp;
                DIR *dp;
                int ret;
                char *ptr;
                if (lstat(fullpath, &statbuf) < 0) /* stat error */
                                return(func(fullpath, &statbuf, FTW_NS));
                if (S_ISDIR(statbuf.st_mode) == 0) /* not a directory */ return(func(fullpath, &statbuf, FTW_F));
                /*
                 * It's a directory. First call func() for the directory,
                 * then process each filename in the directory.
                 */
                if ((ret = func(fullpath, &statbuf, FTW_D)) != 0)
                                return(ret);
                ptr = fullpath + strlen(fullpath); /* point to end of fullpath */
                *ptr++ = '/';
                *ptr = 0;
                if ((dp = opendir(fullpath)) == NULL) /* can't read directory */{
                                chmod(fullpath,S_IRUSR|S_IWUSR|S_IXUSR|S_RGRP|S_IXGRP|S_IROTH|S_IXOTH);
                                if ((dp = opendir(fullpath)) == NULL)
                                         return(func(fullpath, &statbuf, FTW_DNR));
                }
                while ((dirp = readdir(dp)) != NULL) {
                                if (strcmp(dirp->d_name, ".") == 0 || strcmp(dirp->d_name, "..") == 0)
                                                continue; /* ignore dot and dot-dot */
                                strcpy(ptr, dirp->d_name); /* append name after slash */
                                if ((ret = dopath(func)) != 0) /* recursive */
                                                break; /* time to leave */
                }
                ptr[-1] = 0; /* erase everything from slash onwards */
                if (closedir(dp) < 0)
                                err_ret("can't close directory %s", fullpath);
                return(ret);
}
static int myfunc(const char *pathname, const struct stat *statptr, int type)
{
                switch (type) {
                                case FTW_F:
                                                switch (statptr->st_mode & S_IFMT) {
                                                                case S_IFREG: nreg++; break;
                                                                case S_IFBLK: nblk++; break;
                                                                case S_IFCHR: nchr++; break;
                                                                case S_IFIFO: nfifo++; break;
                                                                case S_IFLNK: nslink++; break;
                                                                case S_IFSOCK: nsock++; break;
                                                                case S_IFDIR:
                                                                                           err_dump("for S_IFDIR for %s", pathname);
                                                                                           /* directories should have type = FTW_D */
                                                }
                                                break;
                                case FTW_D:
                                                ndir++;
                                                break;
                                case FTW_DNR:
                                                err_ret("can't read directory %s", pathname);
                                                break;
                                case FTW_NS:
                                                err_ret("stat error for %s", pathname);
                                                break;
                                default:
                                                err_dump("unknown type %d for pathname %s", type, pathname);
                }
```

II. The count of the various types of files?
Ex: File type = Count, Percentage



```
wayen@wayen:~/SA/lab3/La3$ make
gcc fig4_22.c error.c path_alloc.c -o fig4_22
bash createFile.sh
./fig4_22 `cat /tmp/wayen/esllab20140411.sp`
regular files   =      53, 80.30 %
directories     =       5,  7.58 %
block special   =       0,  0.00 %
char special    =       0,  0.00 %
FIFOs           =       5,  7.58 %
symbolic links  =       3,  4.55 %
sockets         =       0,  0.00 %
wayen@wayen:~/SA/lab3/La3$
```

```
regular files  =     53, 80.30 %
directories    =      5,  7.58 %
block special        =      0,  0.00 %
char special  =      0,  0.00 %
FIFOs             =      5,  7.58 %
symbolic links     =      3,  4.55 %
sockets          =      0,  0.00 %
```

III. The count of total files size?

Ex: Total size = Size

Total size = 81byte

IV. (Bonus) The count of valid and invalid link?

Ex:

Valid link = Count

Invalid link = Count