Homework 1

Due Date: January 23, 1998 **Points**: 150

1. (*10 points*) A computer security expert claims that one of the measures necessary to obtain computer security is the separation of programmers and operators. Is she right? Justify your answer.
2. (*20 points*) What is the danger in the following code fragment? How would you fix it?

```
int main(int argc, char *argv[])
{
      char filename[MAXPATHLEN];
      if (argc==1)
            (void) sprintf(filename, "/tmp/xxx%d", getpid());
      else
            (void) sprintf(filename, "/tmp/%s", argv[1]);
      if ((fd = open(filename, O_RDWR|O_EXCL|O_CREAT, 0644) )< 0){
            perror(filename);
            exit(1);
      }
      …
```

3. (*20 points*) Why should the functions *gets*(3), *strcpy*(3), and *strcat*(3) not be used when writing programs? What functions should you use instead?
4. (*100 points*) This problem asks you to write a program to manage UNIX groups. A group is defined by an entry in the file */etc/group* that has the form:

   `groupname:group_password:group_number:membership`

   For example, the line

   `class::1035:bishop,knowles,levitt,fulton`

   defines a group called "class", with no password and GID 1035, and with members bishop, knowles, levitt, and fulton.

   Only the system administrator can manipulate the password or membership of a group. Your task is to write a program that will allow the **first** user named in the membership of the group to:

   a. Change (or set) the group password;

   b. Add a user to the group, or delete a user from the group.

   What follows are some suggestions that may help you.

   i. You do not have privileges to alter the real group file, so make a copy of ~cs153/hw1/group; this has the same format as the system /etc/group file. Please make your group file reading and writing functions modular, so they can be changed to act on the system file by changing one macro. Also, call your group file "group".

   ii. Validate the user's password. So, in the above example, your program would first check that bishop is running it (using *getuid*(2)), and then authenticate bishop.

   iii. You will need to lock the group file so at most one process at a time tries to alter it (why?)