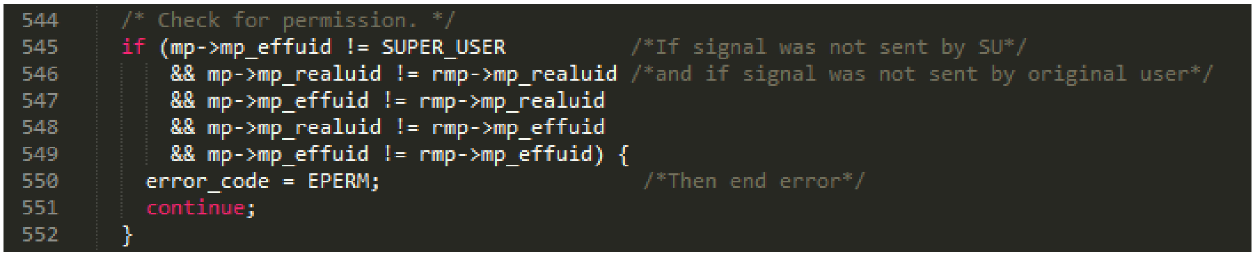
Pablo Arango
Project 1
COP4600
9/19/14

Group Kill() (Modified Kill())

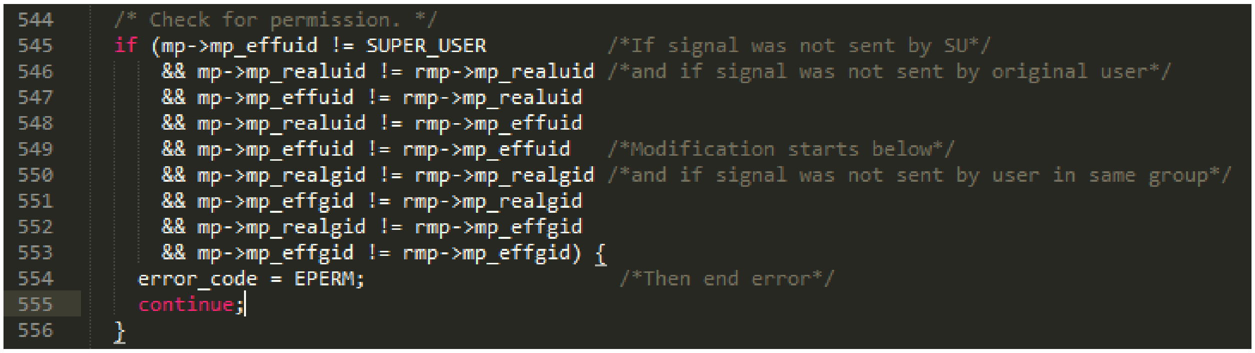
Kill Command Modifications

To allow other members of the same group as the user to send the kill command to programs started by the user, the check for permission part of the signal.c file was modified to not send the error code EPERM when a user with the same effgid or same realgid tried to send the kill signal.



```
/* Check for permission. */
if (mp->mp_effuid != SUPER_USER          /*If signal was not sent by SU*/
    && mp->mp_realuid != rmp->mp_realuid /*and if signal was not sent by original user*/
    && mp->mp_effuid != rmp->mp_realuid
    && mp->mp_realuid != rmp->mp_effuid
    && mp->mp_effuid != rmp->mp_effuid) {
  error_code = EPERM;                     /*Then end error*/
  continue;
}
```

Signal.c BEFORE ^^

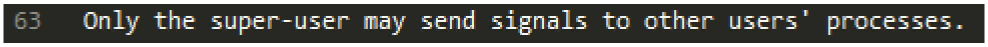


```
/* Check for permission. */
if (mp->mp_effuid != SUPER_USER          /*If signal was not sent by SU*/
    && mp->mp_realuid != rmp->mp_realuid /*and if signal was not sent by original user*/
    && mp->mp_effuid != rmp->mp_realuid
    && mp->mp_realuid != rmp->mp_effuid
    && mp->mp_effuid != rmp->mp_effuid   /*Modification starts below*/
    && mp->mp_realgid != rmp->mp_realgid /*and if signal was not sent by user in same group*/
    && mp->mp_effgid != rmp->mp_realgid
    && mp->mp_realgid != rmp->mp_effgid
    && mp->mp_effgid != rmp->mp_effgid) {
  error_code = EPERM;                     /*Then end error*/
  continue;
}
```

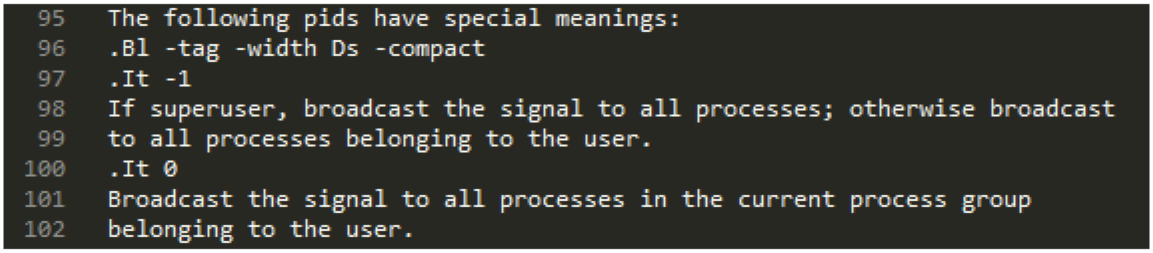
Signal.c AFTER ^^

Kill Manual Modifications

Kill(1)

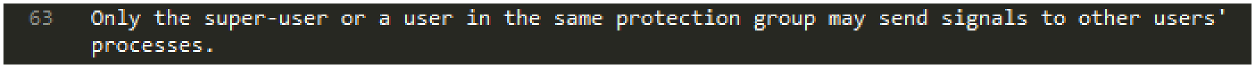


```
Only the super-user may send signals to other users' processes.
```

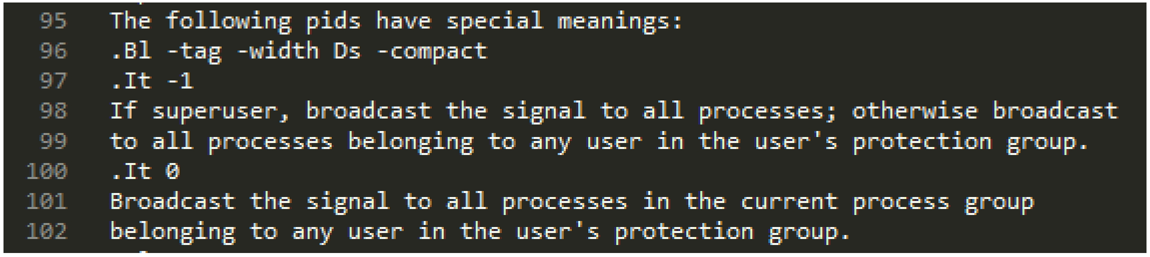


```
The following pids have special meanings:
.Bl -tag -width Ds -compact
.It -1
If superuser, broadcast the signal to all processes; otherwise broadcast
to all processes belonging to the user.
.It 0
Broadcast the signal to all processes in the current process group
belonging to the user.
```

Kill.1 BEFORE ^^



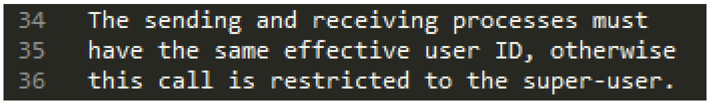
```
Only the super-user or a user in the same protection group may send signals to other users'
processes.
```



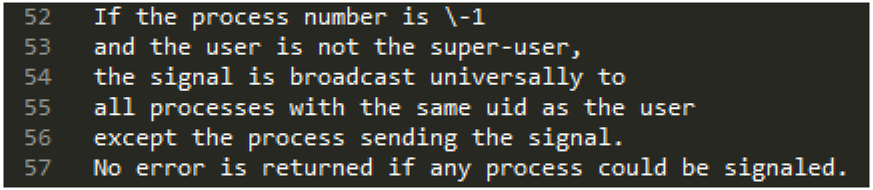
```
The following pids have special meanings:
.Bl -tag -width Ds -compact
.It -1
If superuser, broadcast the signal to all processes; otherwise broadcast
to all processes belonging to any user in the user's protection group.
.It 0
Broadcast the signal to all processes in the current process group
belonging to any user in the user's protection group.
```

Kill.1 AFTER^^

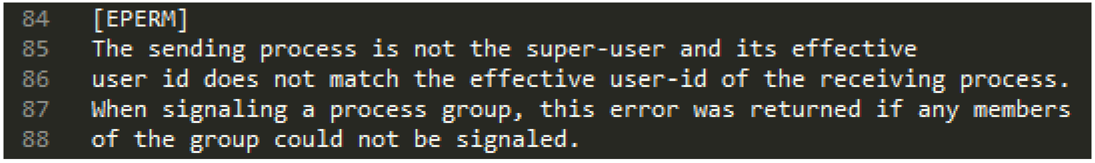
Kill.2



```
The sending and receiving processes must
have the same effective user ID, otherwise
this call is restricted to the super-user.
```

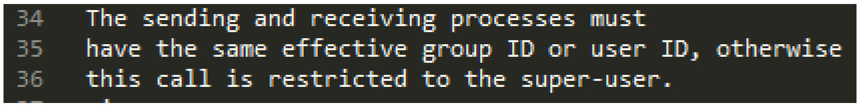


```
If the process number is \-1
and the user is not the super-user,
the signal is broadcast universally to
all processes with the same uid as the user
except the process sending the signal.
No error is returned if any process could be signaled.
```

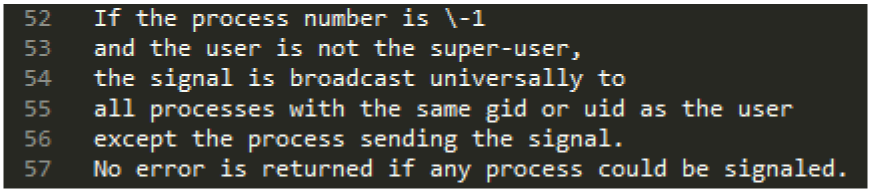


```
[EPERM]
The sending process is not the super-user and its effective
user id does not match the effective user-id of the receiving process.
When signaling a process group, this error was returned if any members
of the group could not be signaled.
```

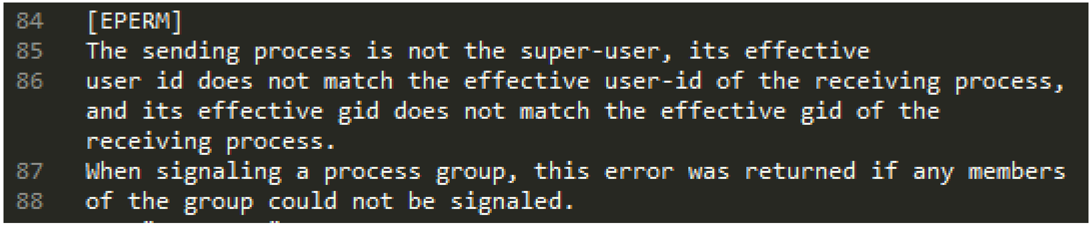
Kill.2 BEFORE^^



```
The sending and receiving processes must
have the same effective group ID or user ID, otherwise
this call is restricted to the super-user.
```



```
If the process number is \-1
and the user is not the super-user,
the signal is broadcast universally to
all processes with the same gid or uid as the user
except the process sending the signal.
No error is returned if any process could be signaled.
```



```
[EPERM]
The sending process is not the super-user, its effective
user id does not match the effective user-id of the receiving process,
and its effective gid does not match the effective gid of the
receiving process.
When signaling a process group, this error was returned if any members
of the group could not be signaled.
```

Kill.2 AFTER^^

Pablo Arango
Project 1
COP4600
9/19/14

Lessons from this project

From this project I learned what effuid, realuid, effgid, and realgid mean. I also learned how the kill call checks who sent the command and if they are allowed to do so. I learned how to mount the files of the minix operating system on my windows machine to edit them using sublime. I learned how to use many commands and what they do in the minix operating system.

Biggest Challenge

The biggest challenge in this project was finding where the system call files where in the directory and how to edit them when the system was running on a virtual machine. To find them and edit them I had to learn how to mount the operating system files as a drive on my windows machine using a sftp file system.