System Programming – Project 2
(2015-2016 Fall Term)

In this project you are asked to add a new field to the task descriptor. The name and type of the field is:

```
int casper;
```

- If `casper=0` : The process is visible to all, i.e. it is listed in the `/proc` file system and it can be seen using "`ps`", "`pstree`", "`top`", …
- If `casper=1` : The process is visible only to processes which have the same user id, i.e. for all other processes, it is NOT listed in the `/proc` file system and it can NOT be seen using "`ps`", "`pstree`", "`top`", …
- If `casper=2` : The process is visible only to processes which are in the same group, i.e. for all other processes, it is NOT listed in the `/proc` file system and it can NOT be seen using "`ps`", "`pstree`", "`top`", …
- If `casper=3` : The process is invisible for all, i.e. it is NOT listed in the `/proc` file system and it can NOT be seen using "`ps`", "`pstree`", "`top`", …

For this project, you are also required to write a system call which sets the value of the `casper` flag in the task descriptor of a process. The prototype for the system call will be

```
long set_casper(pid_t pid, int value);
```

where `pid` is the process id of the process whose `casper` flag is set and `value` shows the number to be assigned to the `casper` flag. `value` can be `0, 1, 2,` or `3.` Any other value causes an error.

Only processes having root privileges can successfully execute this system call.

On error, the `set_casper` system call returns an appropriate error message. Otherwise, it returns `0`.
(Note: To see a list of default error codes, refer to the manual pages using "man errno".)

For the project you need to,

1. Add a new field to the task descriptor. This field should be added to the end of the task descriptor. The name and type of the field is:
   ```
   int casper;
   ```
2. Modify the code used by the kernel when creating and initializing new processes. A newly created process should have its `casper` field initialized to `0`. (Note: Remember how the process with `pid=0` is created and initialized in Linux.)
3. Write a system call which changes the value of the `casper` field in the task descriptor if the caller process has root privileges. Add your system call to the kernel.

4. Modify the code that generates the `/proc` filesystem so that its behavior is according to the value of the `casper` flag..
5. Write a short test program that accepts the pid of the process and the flag value as input and makes the `set_casper` system call. The test program should output the return value of the system call. Experiment by running the program with and without root privileges.

Please read the following carefully!

- You are required to submit the following files through the Ninova system as a zip file:
  - Source codes of all kernel code files you modify,
  - Source codes of your test programs,
  - A text file listing only the changed parts in the code files you modified (Hint: Use the diff command to compare files line by line).
- Each member of the group must make a submission, even though the submitted files may be the same for all group members.
- Group members will be graded individually based on their performance in the lab session and the submitted group project. Students who are not present during the lab session will not receive a grade for the project, even though they may have made a submission through the Ninova system.
- Any form of cheating or plagiarism will not be tolerated. The submitted work should be the product of the group itself; collaboration or code sharing between different groups will be accepted as plagiarism. This also includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (even when attributed). Serious offences will be reported to the administration for disciplinary measures.