Assignment 7

(1) **(1 pt)** Fork a new child.

Answer: A new child is created on line 10:

```c
// Fork a new child
pid = fork();
```

(2) **(1 pt)** Create a new session in the child process.

Answer: We create a new session in the child process using `setsid()`:

```c
// Create a new session
if (setsid() < 0) {
  perror("setsid failed");
  exit(1);
}
```

(3) **(1 pt)** Verify that the child is a process group leader and no longer has a controlling terminal by using ps command to print *PID*, *PGRP*, and *TPGID*.

Answer: We call `ps` from the parent process, using the `-o` option for the `ps` command to display the required columns, and `-p` to specify our child process.

```c
// Use ps(1) command within the program to get info.
sprintf(command, "ps -o pid -o pgid=PGRP -o tpgid -p %d", pid);
```

using the form `keyword=NAME`, we can customize the column names shown in the output. Then we can read the output of that command to display to stdout.

```c
FILE *ps_output = popen(command, "r");
if (ps_output == NULL) {
  perror("popen failed");
  exit(EXIT_FAILURE);
}

// print the output of the ps(1) command
char buf[1024];
while (fgets(buf, sizeof(buf), ps_output) != NULL) {
  printf("%s",buf);
}
```

(4) **(1 pts)** Explain why the child process does not have a controlling terminal. What are *PID*, *PGRP*, and *TPGID* values in this case? What are their meanings? Please answer these questions in your report.

Answer: The child process does not have a controlling terminal since it is in a new session with no controling terminal formed. We get the following output from the program:

```
  PID  PGRP TPGID
72092 72092     0
```

Here, `PID` and `PGRP` are the same , and `TPGID` is zero, indicating the absence of a controlling terminal.

- `PID` is the process id, confirming that we are viewing the child process.
- `PGID` is the process group id. This is the same as the `PID` since group IDs are usually set to the `PID` of the process that created the group.
- `TPGID` is the control terminal process group ID, which signifies the group that that controlling terminal belongs to. `TPGID` is set when the session leader (in this case our child process) establishes a connection to a controling terminal. Since that has not been done, this value is zero.