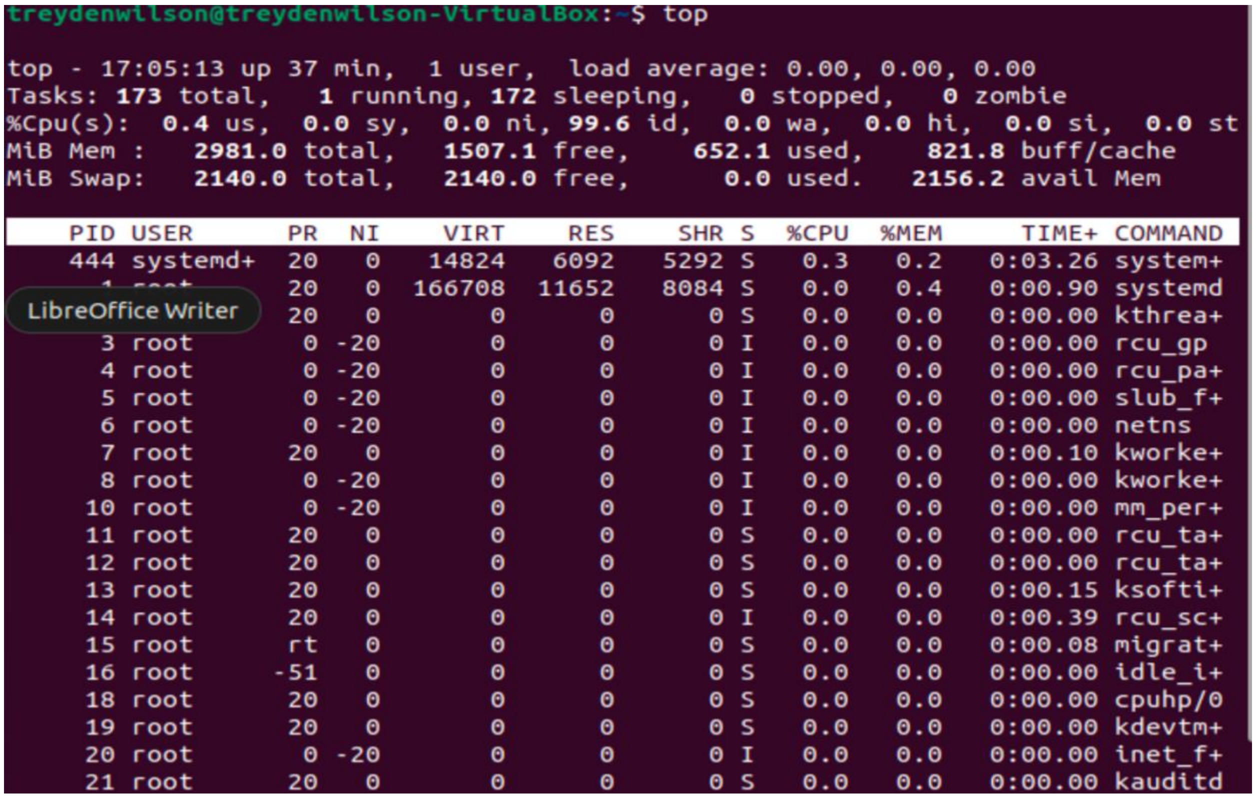
Treyden Wilson
Lab 3
CS 470 Operating Systems

1. How many child processes are created upon execution of this program?

   There are 3 child processes created upon execution of the given program. Using 2^n-1, where n = 2, we get 4-1 = 3. Hence, 3 child processes are created upon execution.

2. When you start a browser, you will notice the browser process appear in the top display. What does it consume?

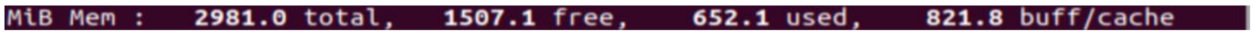
   It consumes 0.4 CPU %, 652.2 memory used and 821.7 buff/cache, 0.0 virtual memory used. 173 total tasks with 1 running.



```
treydenwilson@treydenwilson-VirtualBox:~$ top

top - 17:05:13 up 37 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 173 total,   1 running, 172 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.4 us,  0.0 sy,  0.0 ni, 99.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   2981.0 total,   1507.1 free,    652.1 used,    821.8 buff/cache
MiB Swap:   2140.0 total,   2140.0 free,      0.0 used.   2156.2 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    444 systemd+  20   0   14824   6092   5292 S   0.3   0.2   0:03.26 system+
                  20   0  166708  11652   8084 S   0.0   0.4   0:00.90 systemd
                  20   0       0      0      0 S   0.0   0.0   0:00.00 kthrea+
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_pa+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_f+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.10 kworke+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworke+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_per+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_ta+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_ta+
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.15 ksofti+
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.39 rcu_sc+
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.08 migrat+
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_i+
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtm+
     20 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 inet_f+
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
```

3. How much memory is available in the system?

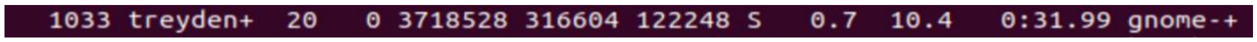
   The memory available in the system is 2981. However, only 1507.1 is free.



```
MiB Mem :   2981.0 total,   1507.1 free,    652.1 used,    821.8 buff/cache
```

4. Which process consumes the most CPU?

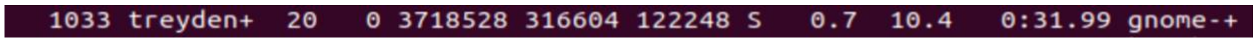
   The process that consumes the most CPU is PID 1033, and it consumes 0.7% of the CPU.



```
   1033 treyden+  20   0 3718528 316604 122248 S   0.7  10.4   0:31.99 gnome-+
```

5. Which process has the most memory?

   The process that has the most memory and memory usage is PID 1033. It consumed 10.4% of the memory and has a virtual memory size of 3718528.



```
   1033 treyden+  20   0 3718528 316604 122248 S   0.7  10.4   0:31.99 gnome-+
```

6. Could you please explain the following commands?
7. apt-get, yum, wget, gzip, tar, rar

   apt-get is a Linux command line tool for managing packages, meaning it can be used to install, update, and remove software packages, along with upgrading the entire system.

   yum is a command line tool for managing packages on Linux systems as well, and functions similarly to apt-get.

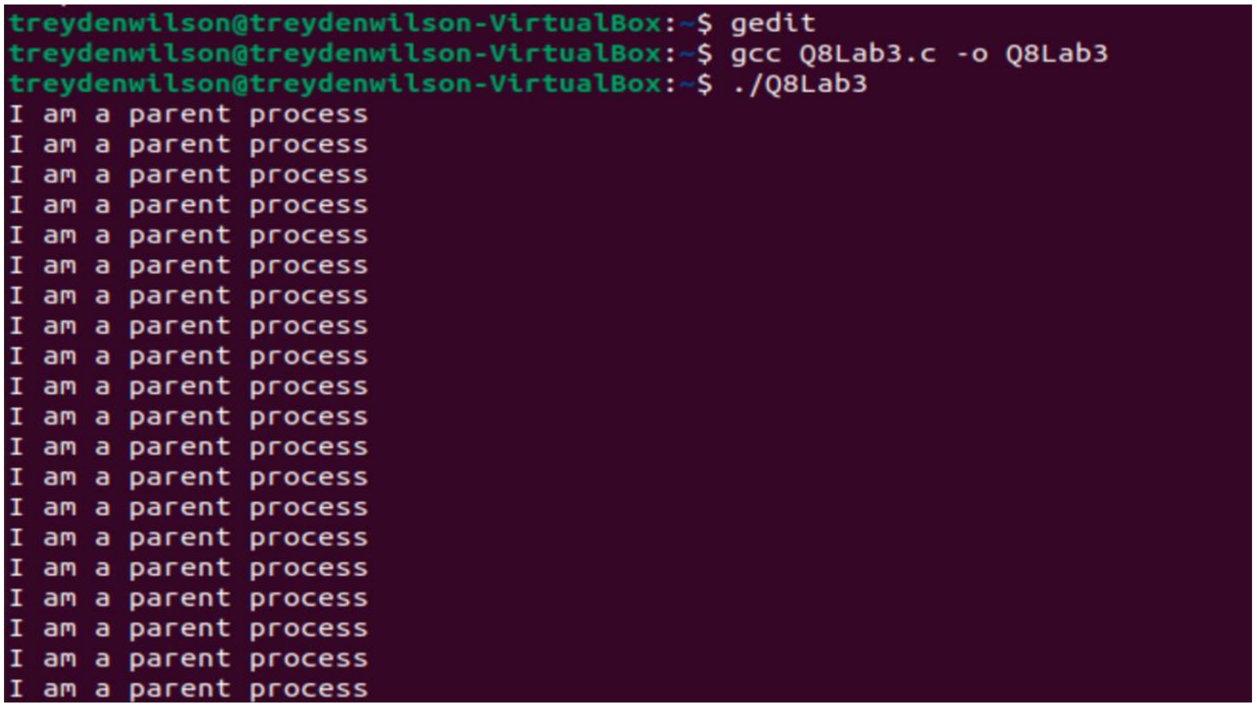
   wget is a command line tool for downloading files from the internet and can be used to download files from a specified URL and save them to the local file system.

   gzip is a command line tool for compressing and decompressing files, where it uses a GZIP algorithm to compress files and reduces the size of files for faster transfers or to save space on the disk.

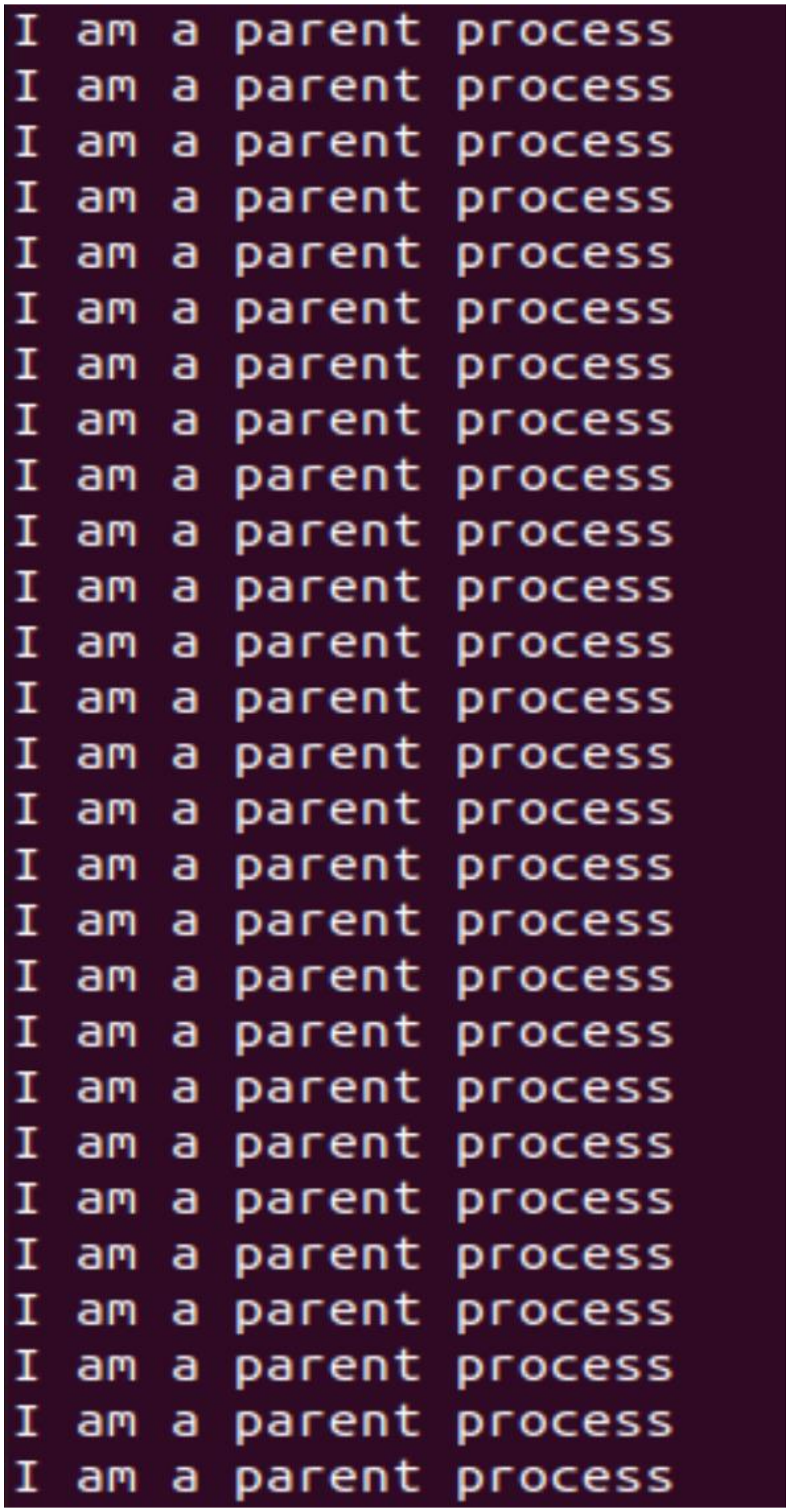
tar is a command line tool for creating and extracting archives and can be used to combine multiple files into a single archive, along with extracting files from an archive.

rar is a command line tool for creating and extracting RAR archives and is not a proprietary software, meaning it may need to be separately installed.

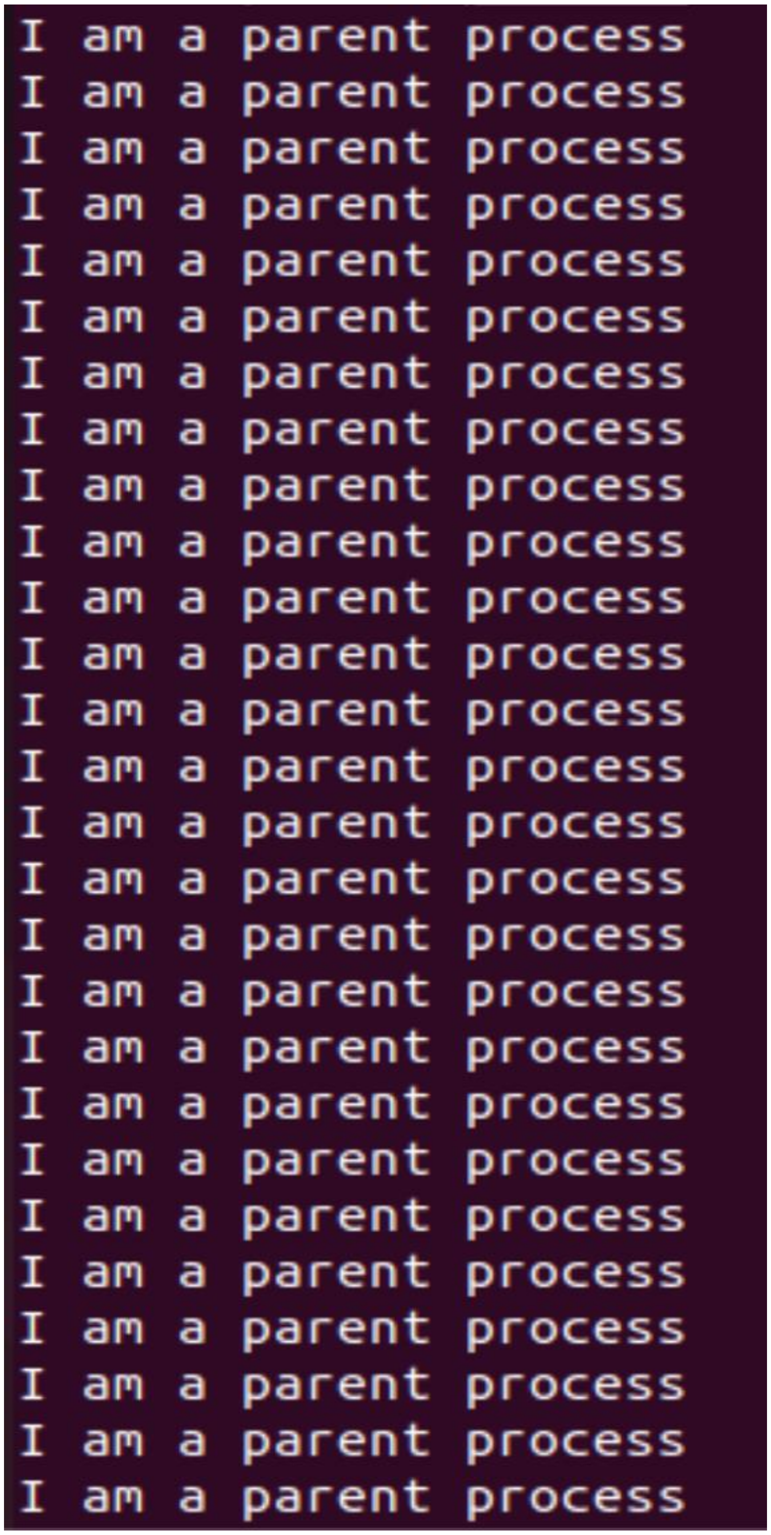
8. Write a program that will generate a child process. In a loop, the child process writes "I am a child process" 200 times and the parent process repeatedly prints "I am a parent process" in a loop.



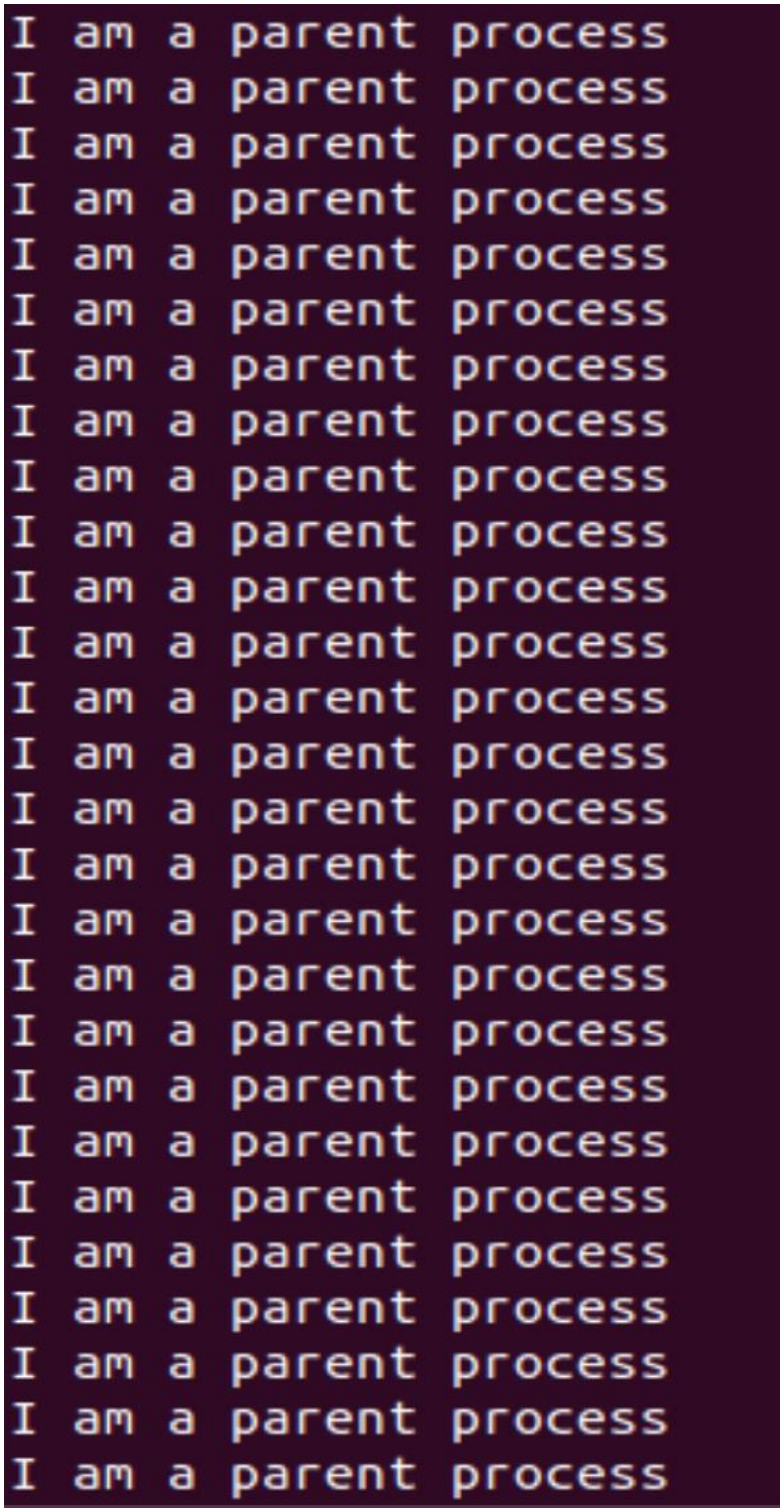
```
treydenwilson@treydenwilson-VirtualBox:~$ gedit
treydenwilson@treydenwilson-VirtualBox:~$ gcc Q8Lab3.c -o Q8Lab3
treydenwilson@treydenwilson-VirtualBox:~$ ./Q8Lab3
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```

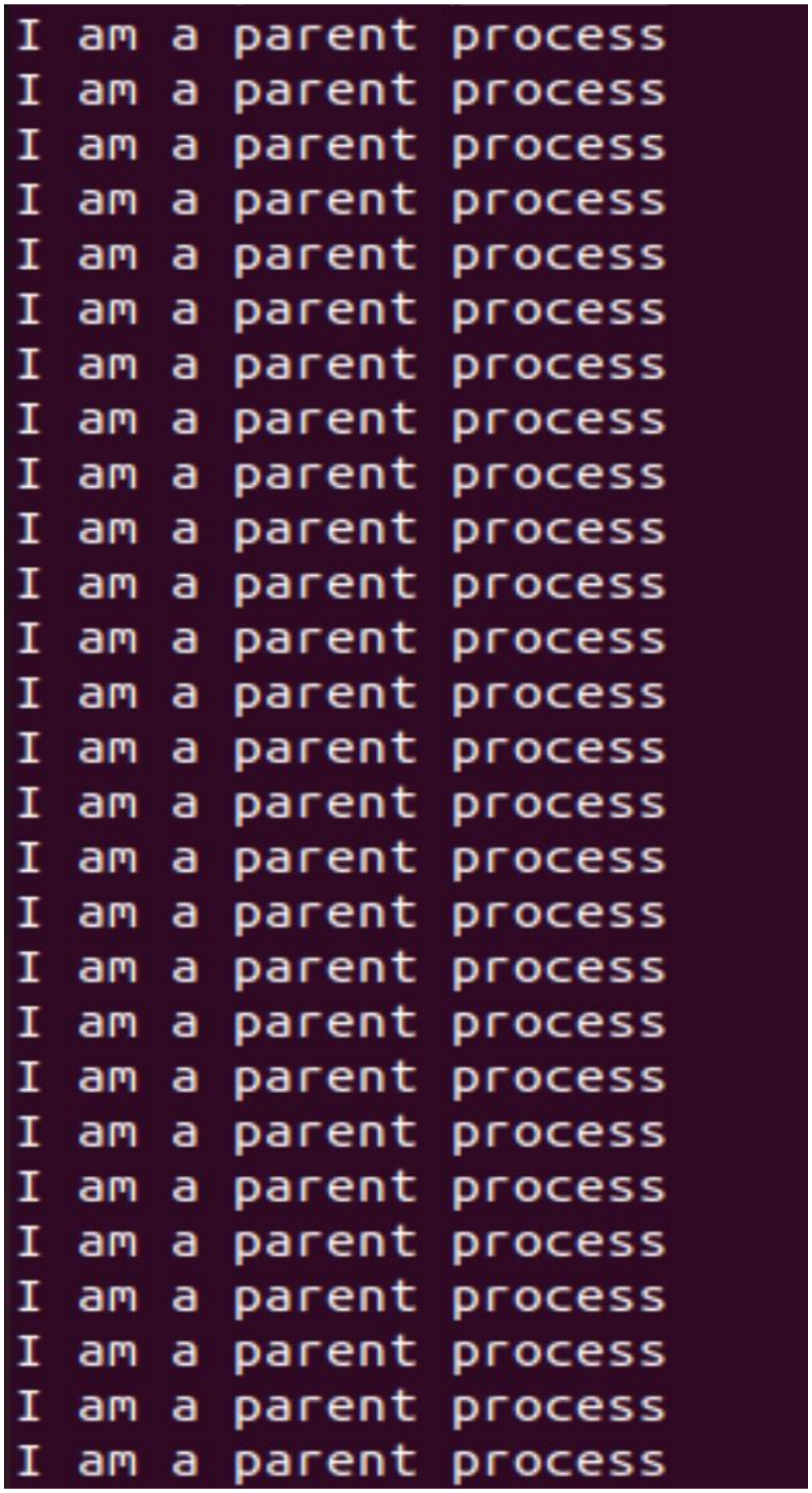


```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```

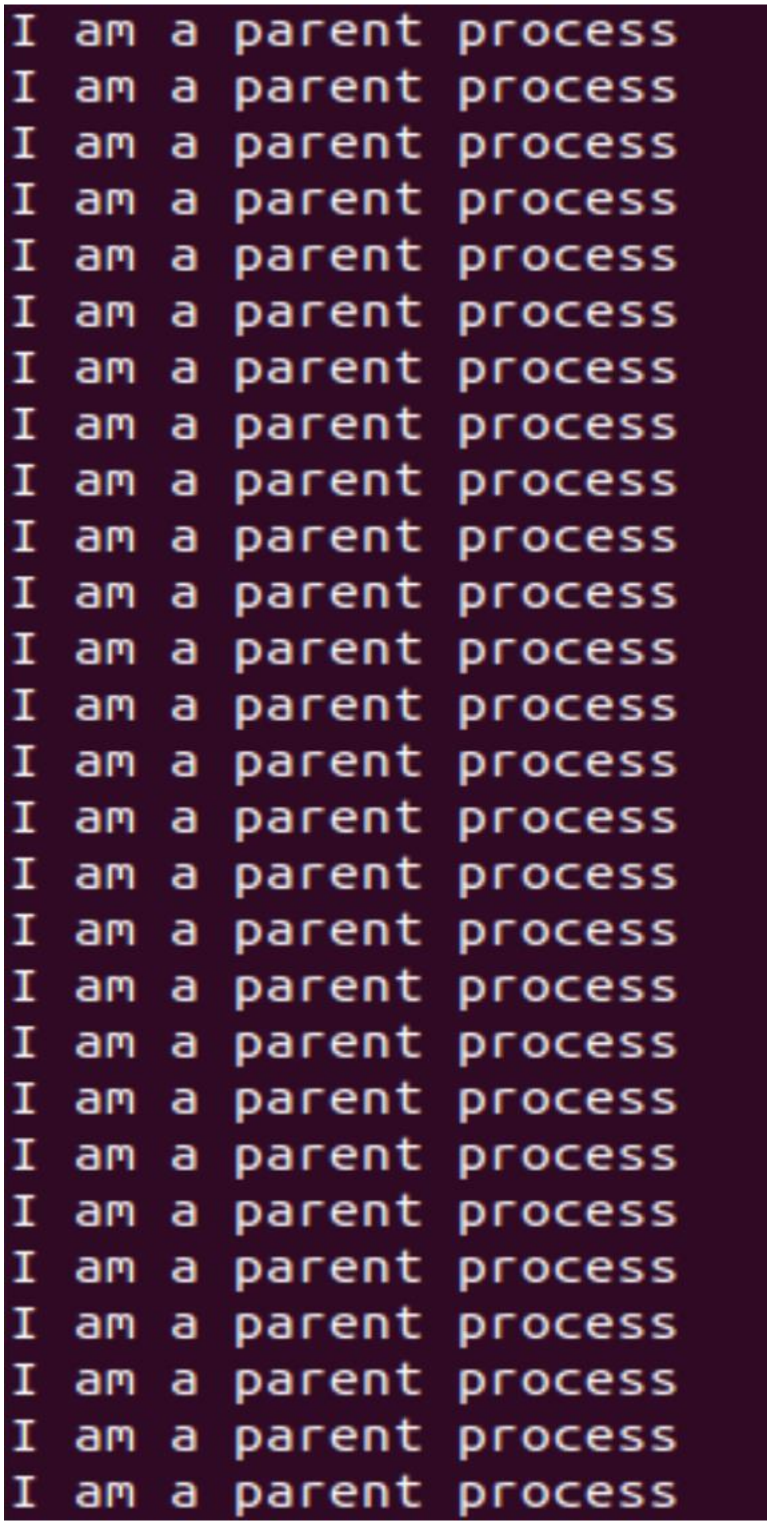


```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```

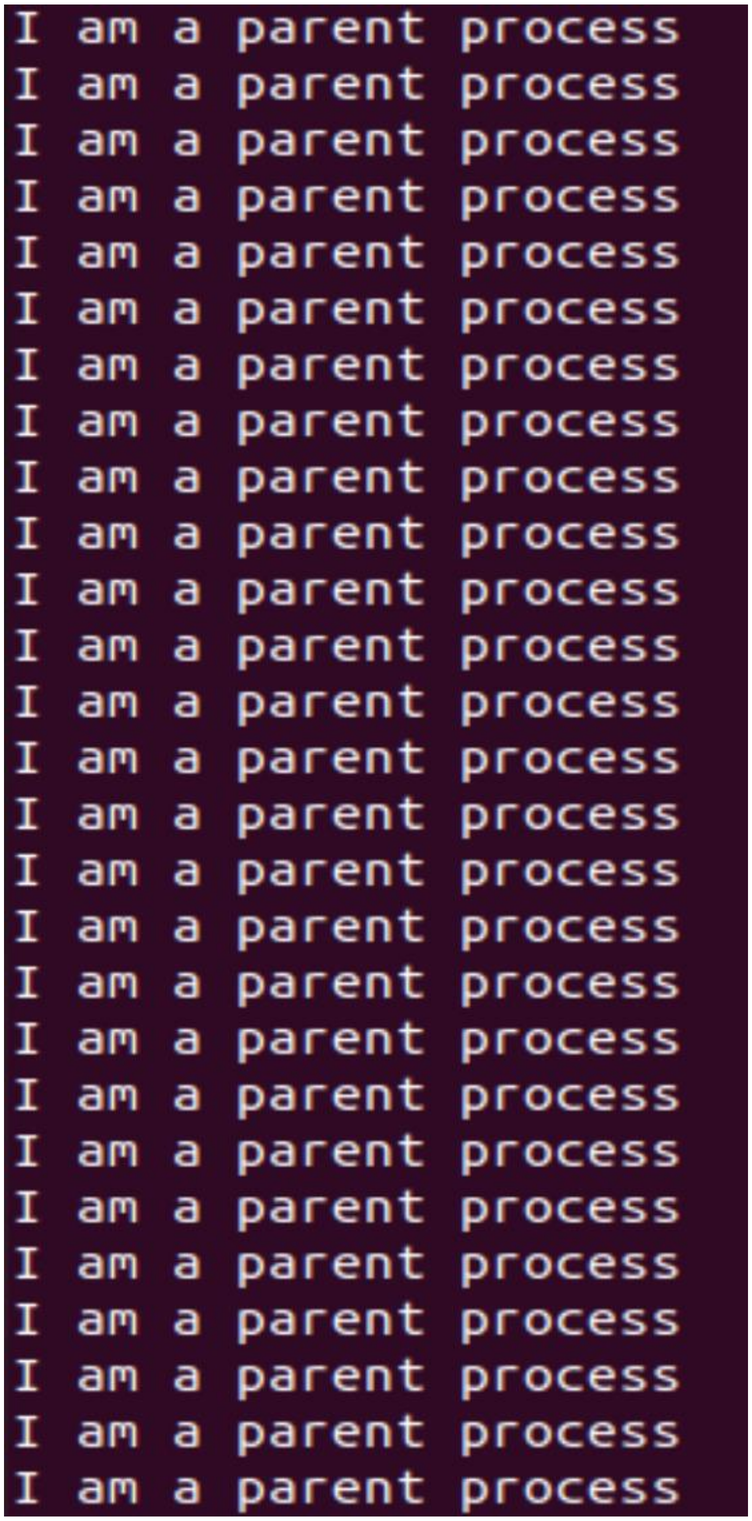

```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```

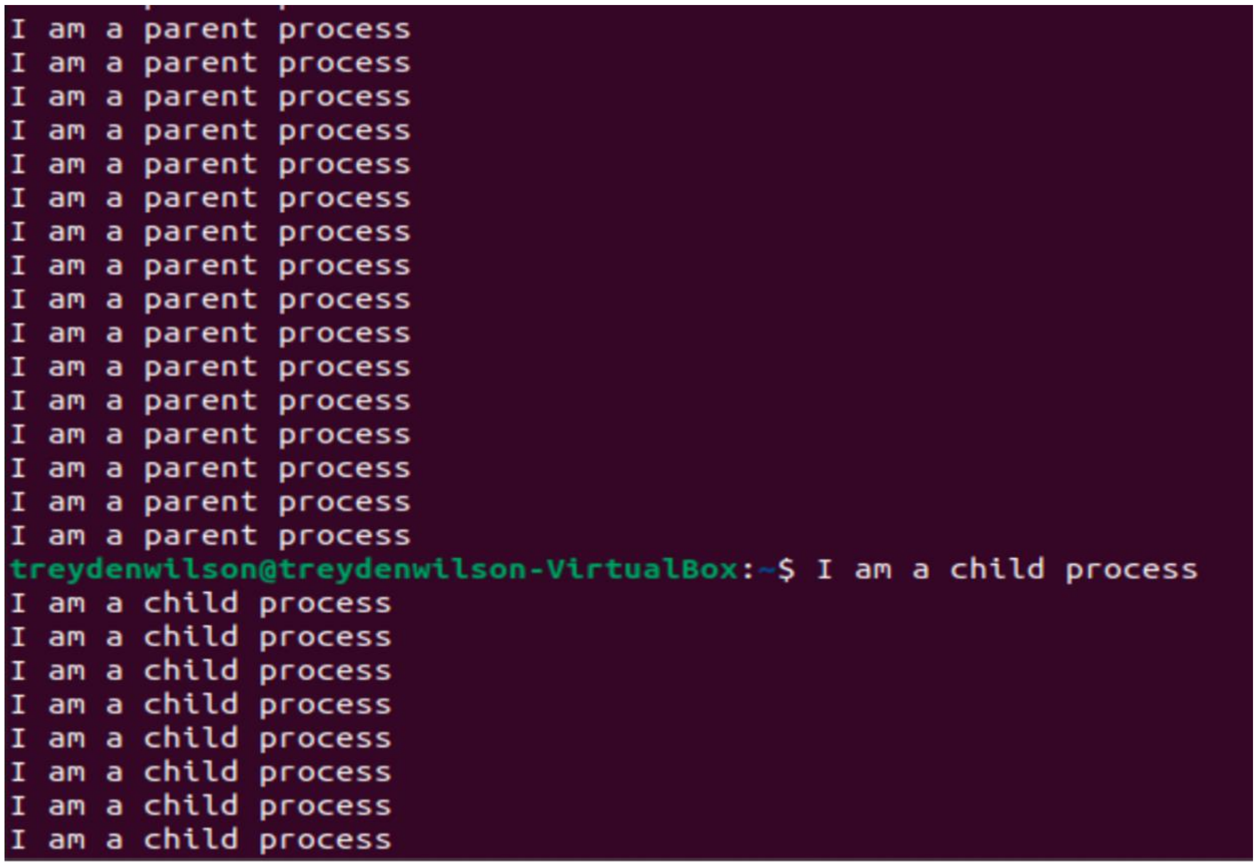


```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```

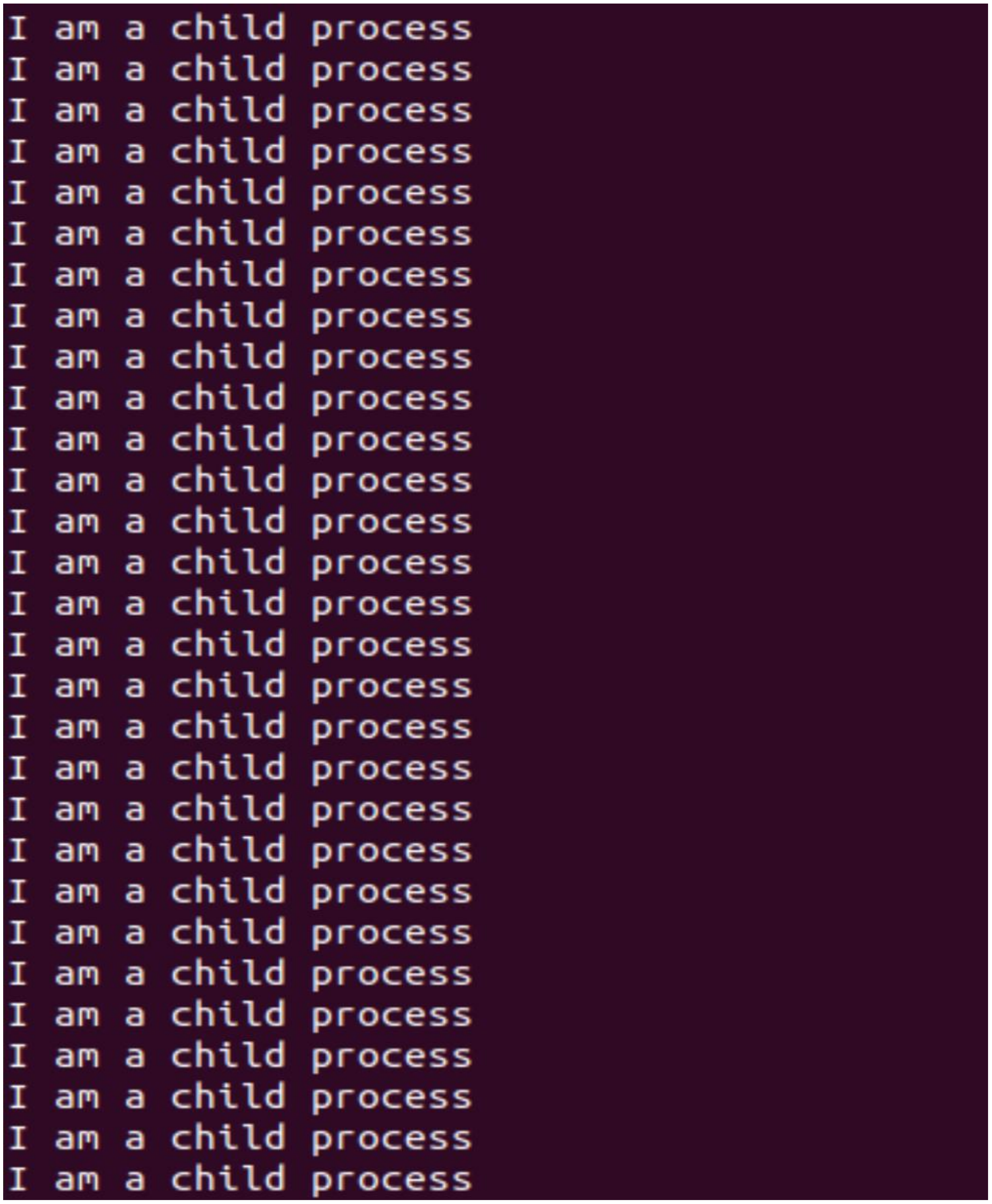


```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```

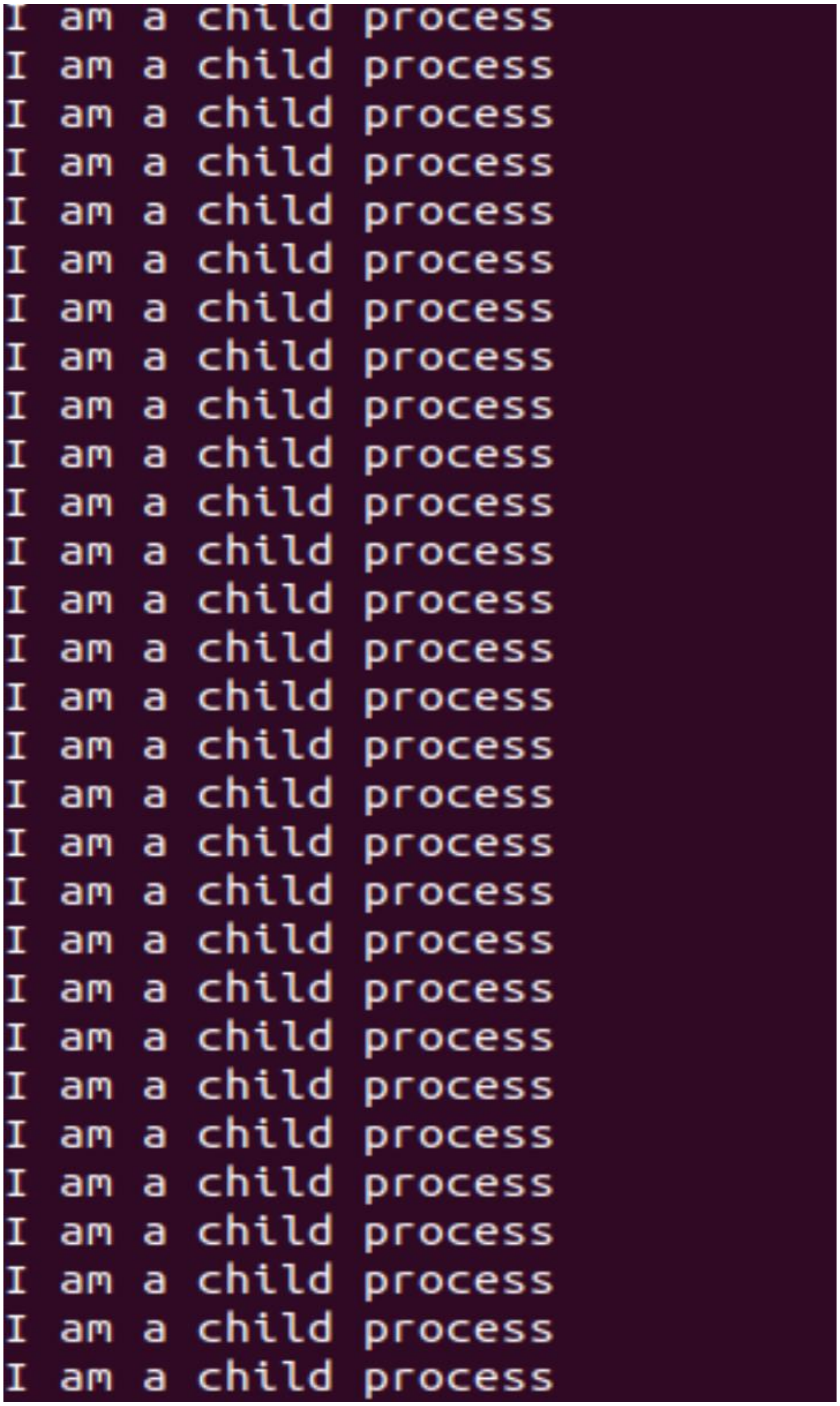

```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
```



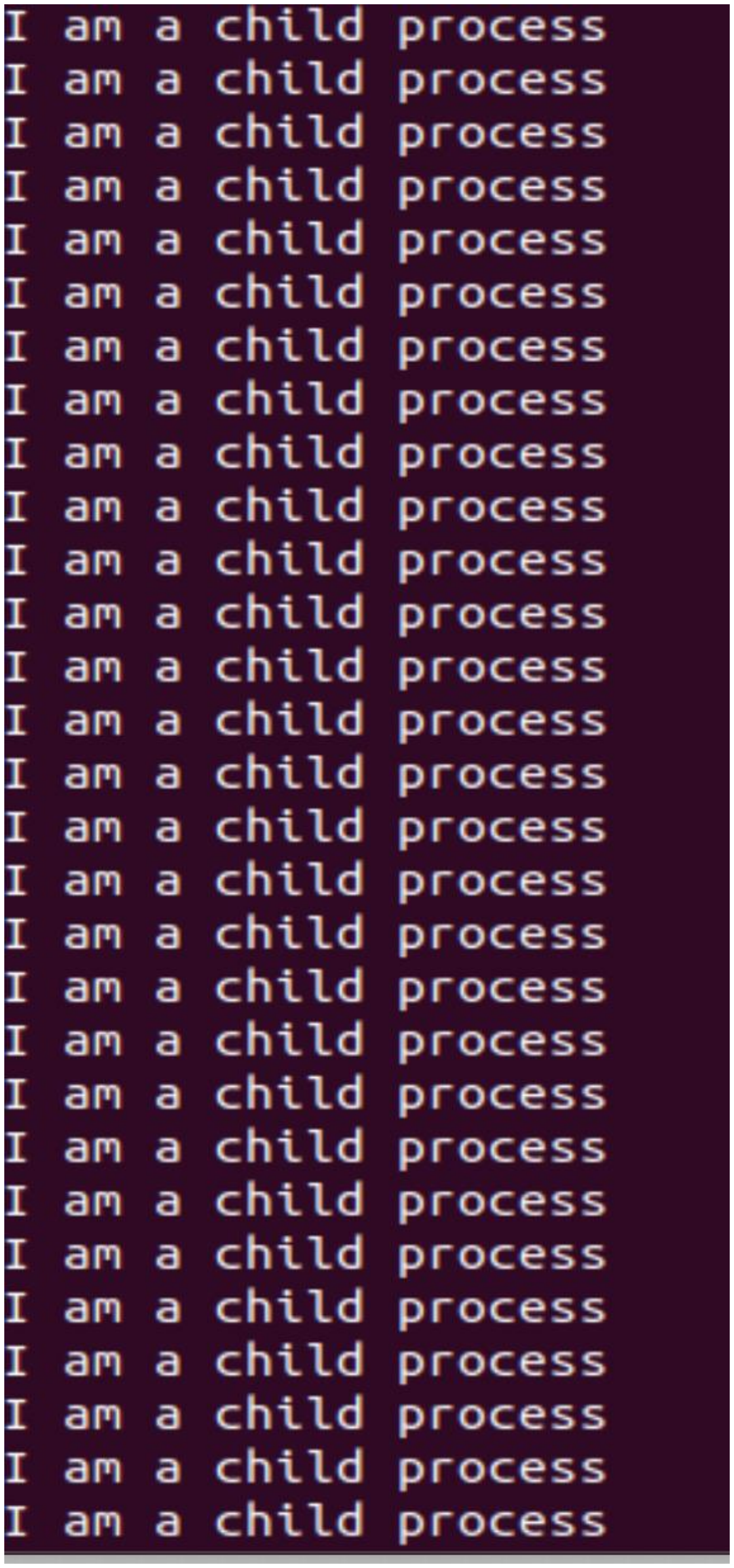
```
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
I am a parent process
treydenwilson@treydenwilson-VirtualBox:~$ I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
```



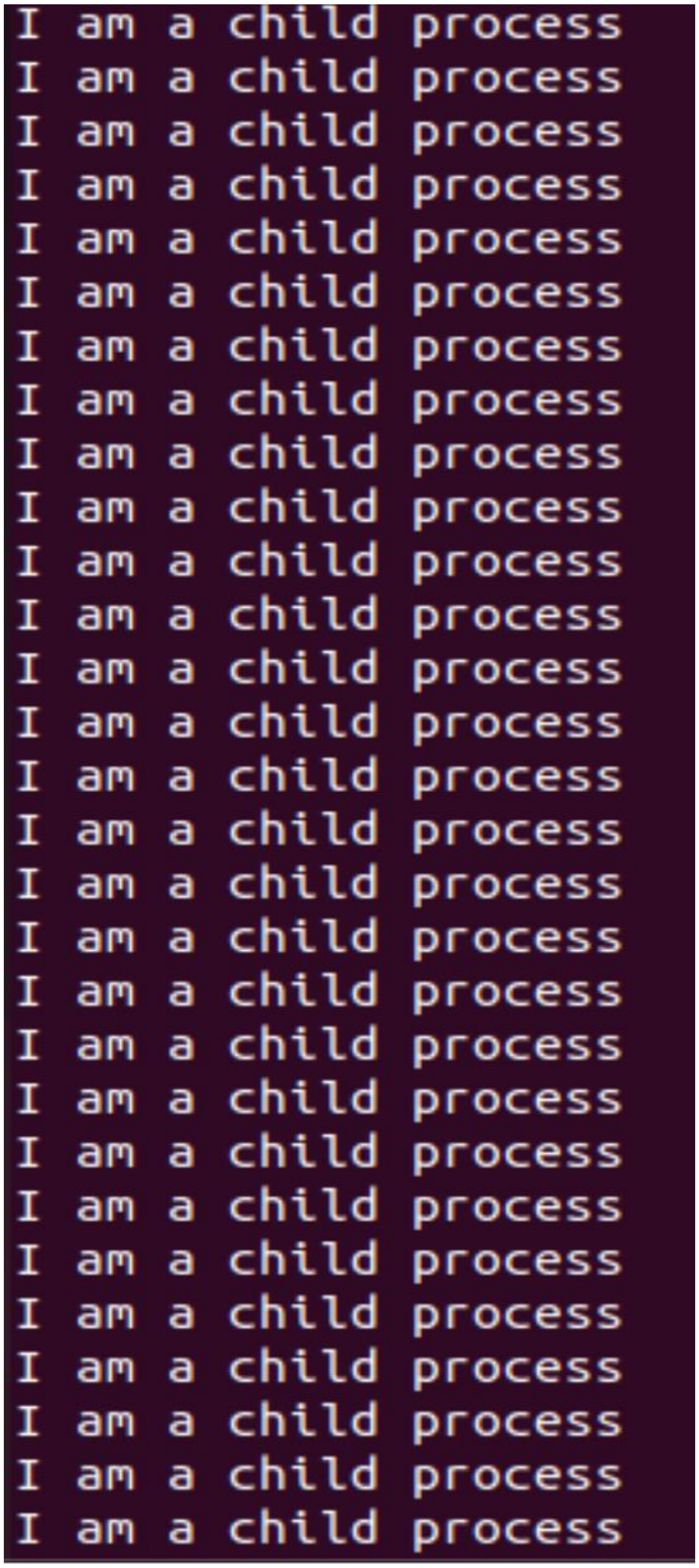
```
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
```



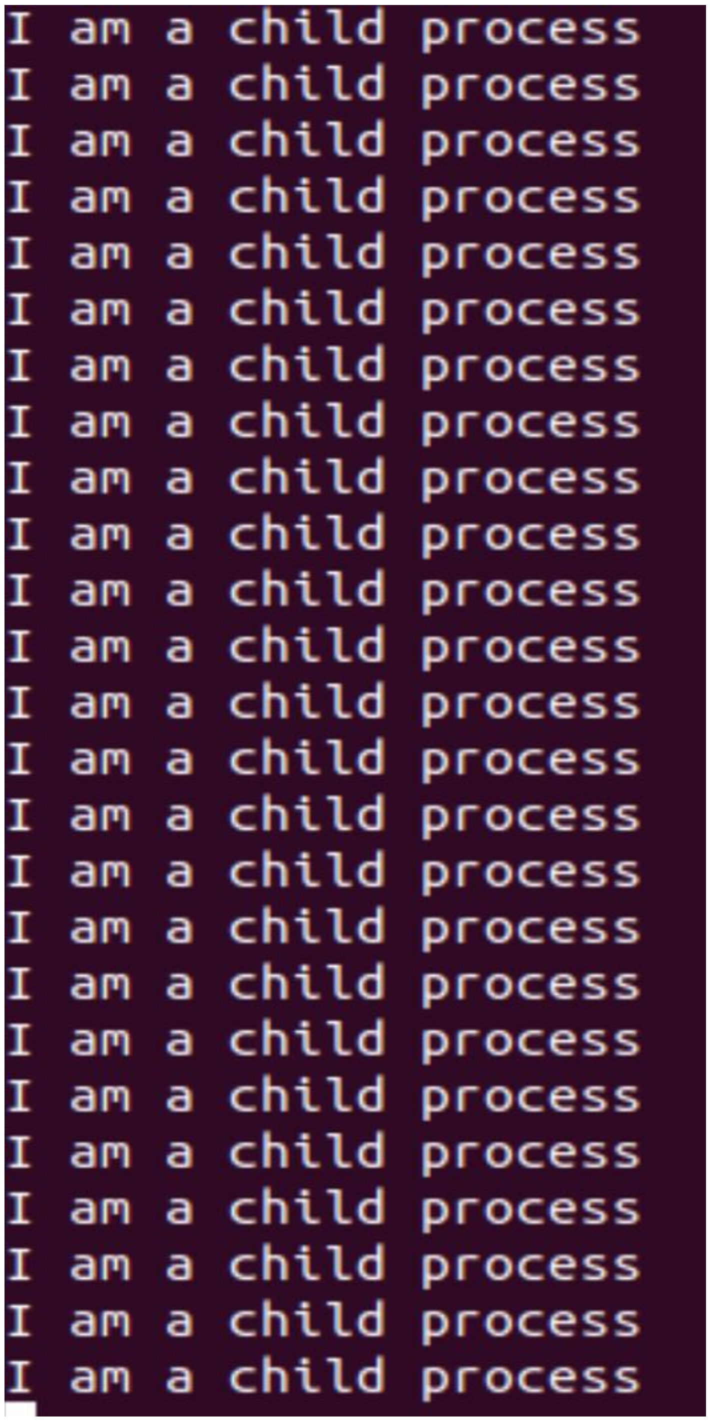
```
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
```



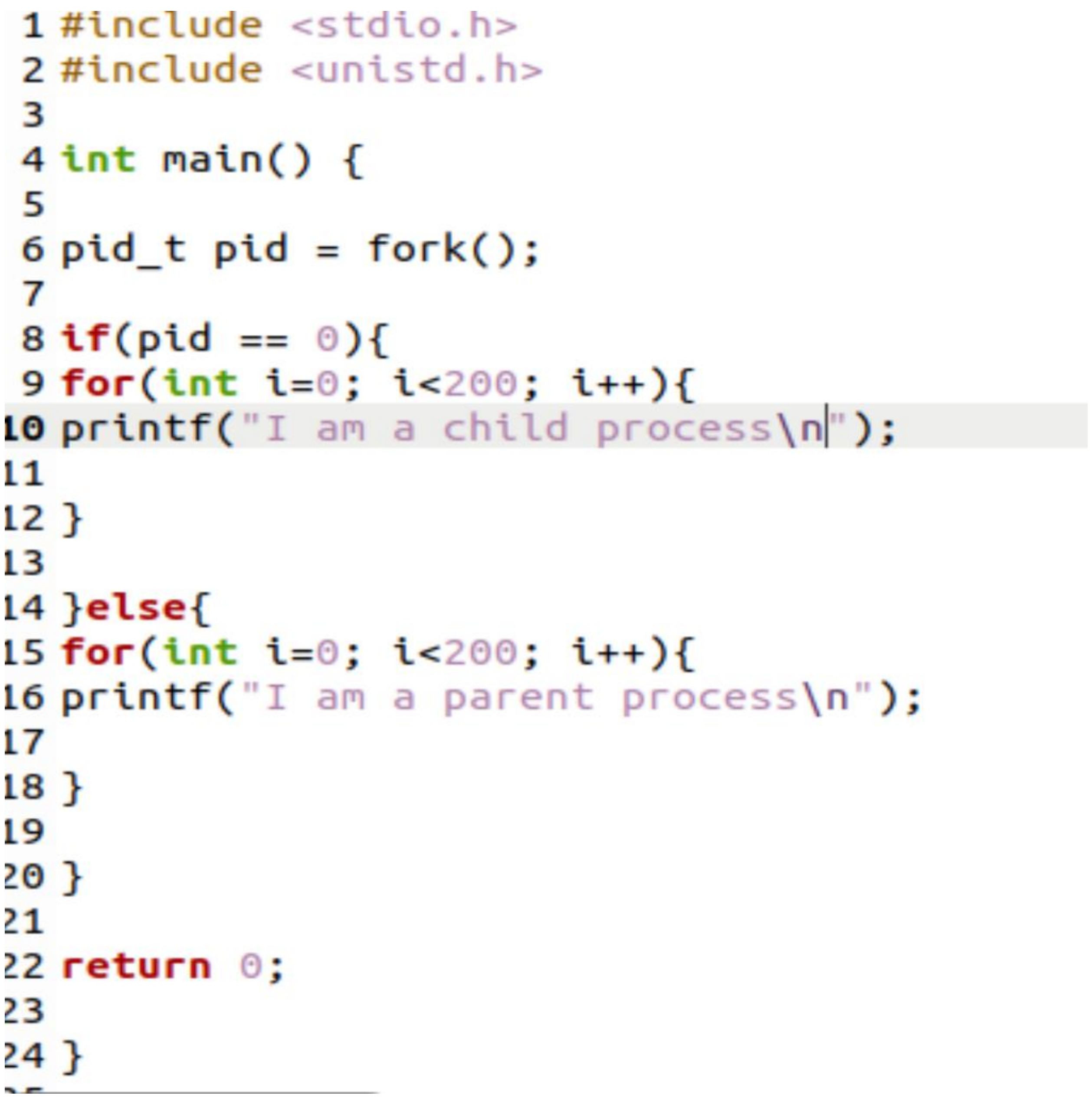
```
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
```



```
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
```

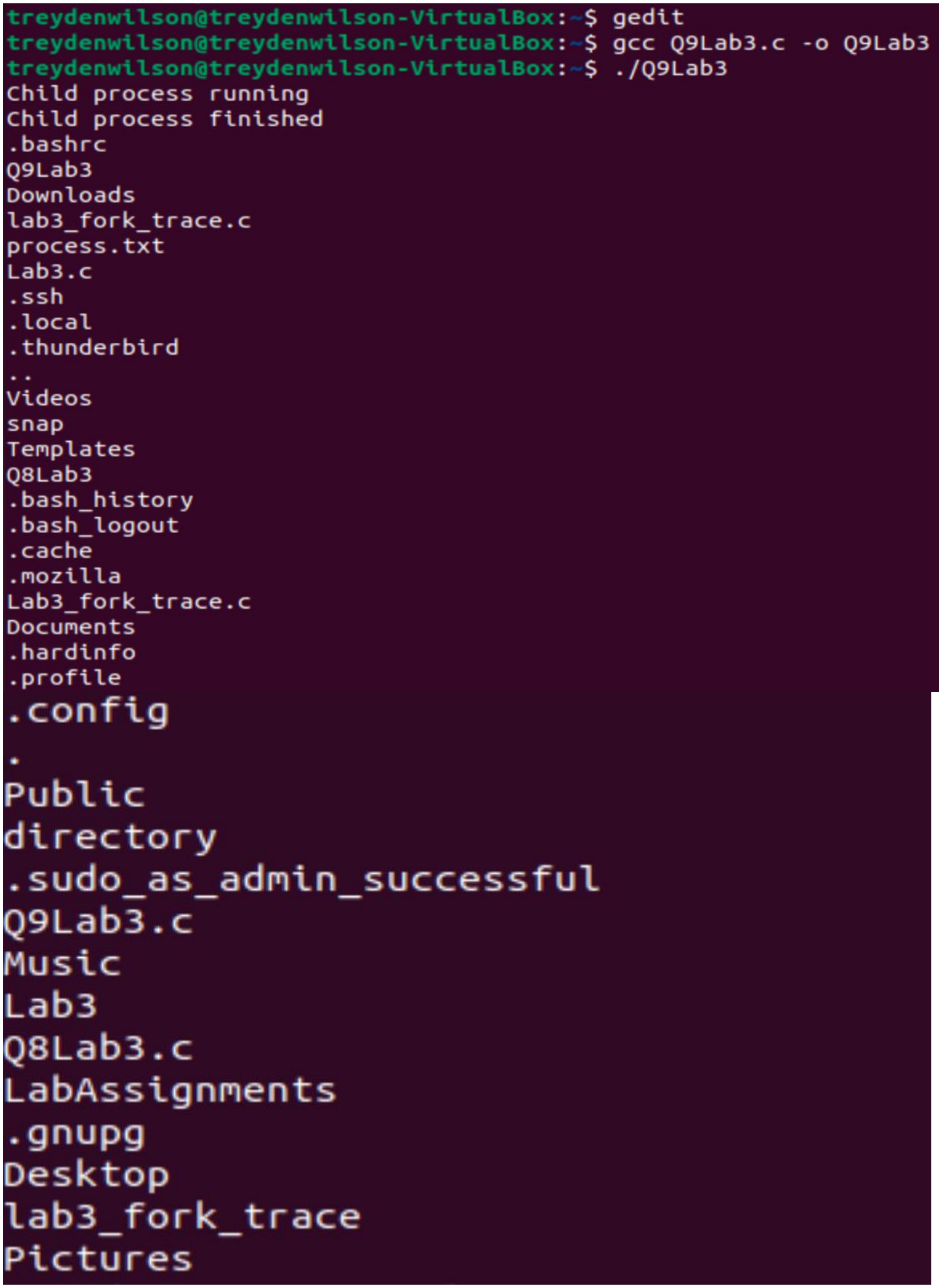


```
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
I am a child process
```

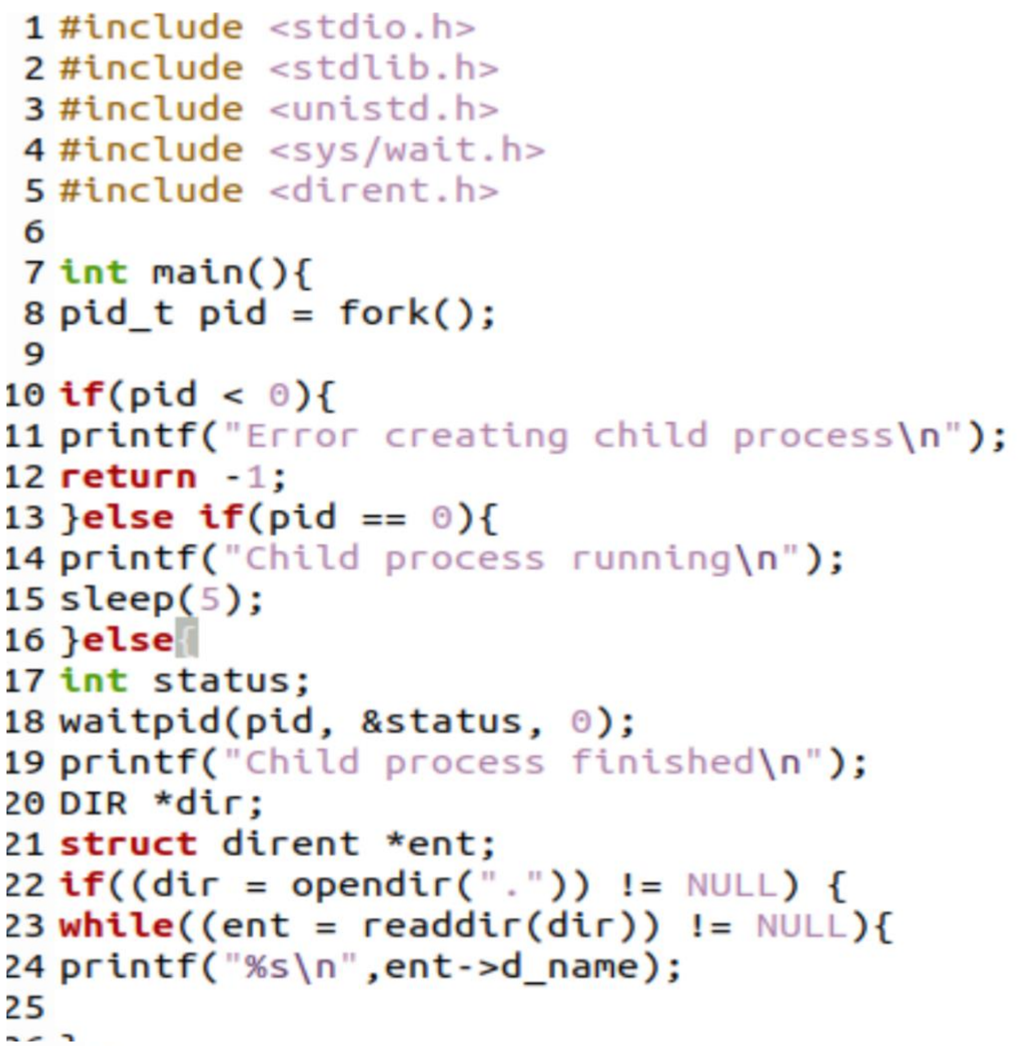
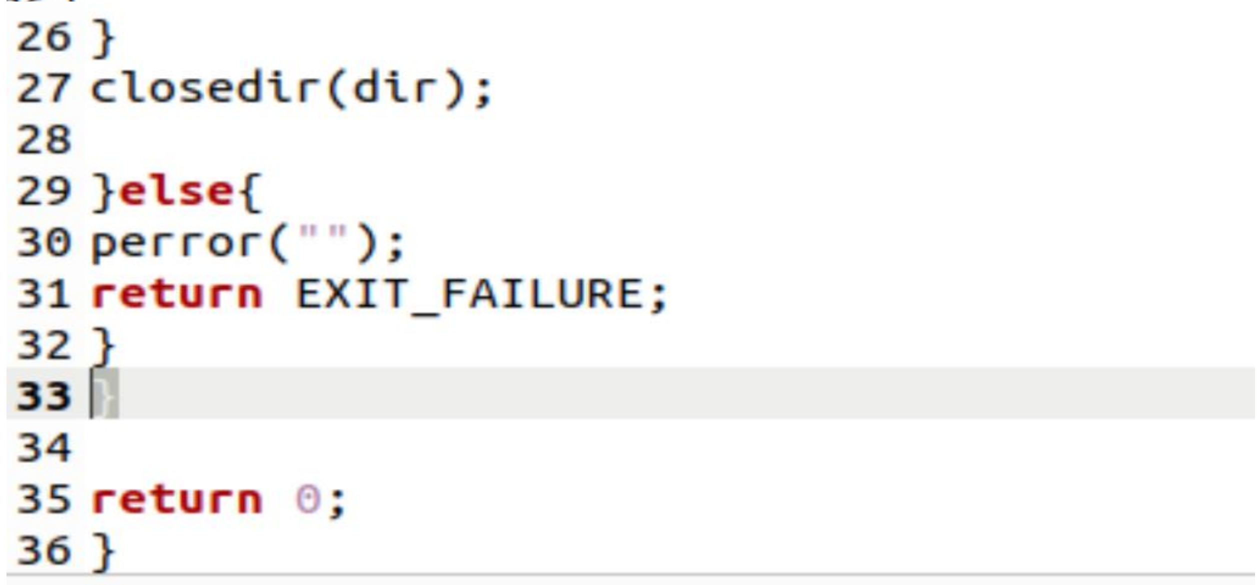


```
#include <stdio.h>
#include <unistd.h>

int main() {

pid_t pid = fork();

if(pid == 0){
for(int i=0; i<200; i++){
printf("I am a child process\n");

}

}else{
for(int i=0; i<200; i++){
printf("I am a parent process\n");

}

}

return 0;

}
```

9. Write a program that create a child process with the fork () system call. The parent process waits for the child process to finish before printing the contents of the current directory.

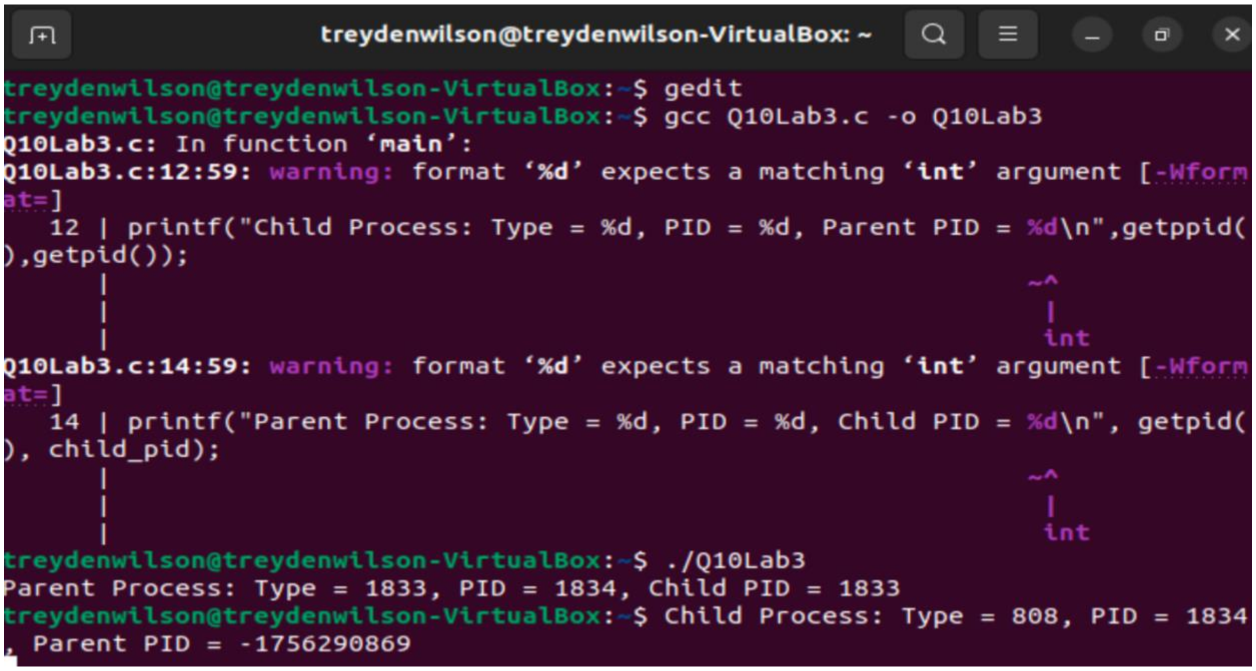


```
treydenwilson@treydenwilson-VirtualBox:~$ gedit
treydenwilson@treydenwilson-VirtualBox:~$ gcc Q9Lab3.c -o Q9Lab3
treydenwilson@treydenwilson-VirtualBox:~$ ./Q9Lab3
Child process running
Child process finished
.bashrc
Q9Lab3
Downloads
lab3_fork_trace.c
process.txt
Lab3.c
.ssh
.local
.thunderbird
..
Videos
snap
Templates
Q8Lab3
.bash_history
.bash_logout
.cache
.mozilla
Lab3_fork_trace.c
Documents
.hardinfo
.profile
.config
.
Public
directory
.sudo_as_admin_successful
Q9Lab3.c
Music
Lab3
Q8Lab3.c
LabAssignments
.gnupg
Desktop
lab3_fork_trace
Pictures
```

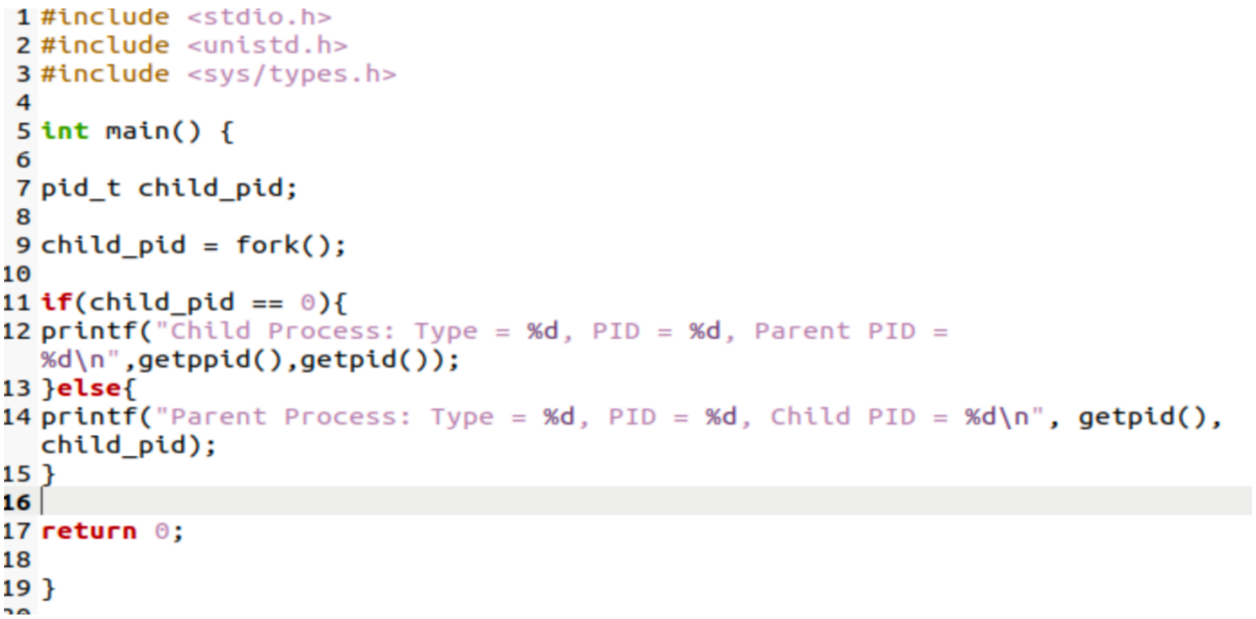



```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#include <dirent.h>

int main(){
pid_t pid = fork();

if(pid < 0){
printf("Error creating child process\n");
return -1;
}else if(pid == 0){
printf("Child process running\n");
sleep(5);
}else{
int status;
waitpid(pid, &status, 0);
printf("Child process finished\n");
DIR *dir;
struct dirent *ent;
if((dir = opendir(".")) != NULL) {
while((ent = readdir(dir)) != NULL){
printf("%s\n",ent->d_name);

}
closedir(dir);

}else{
perror("");
return EXIT_FAILURE;
}
}

return 0;
}
```

10. Write a program that create a child process with the fork () system call and print its PID. Following a fork () system call, both parent and child processes print their process type and PID. Additionally, the parent process prints the PID of its child, and the child process prints the PID of its parent.



```
treydenwilson@treydenwilson-VirtualBox:~$ gedit
treydenwilson@treydenwilson-VirtualBox:~$ gcc Q10Lab3.c -o Q10Lab3
Q10Lab3.c: In function ‘main’:
Q10Lab3.c:12:59: warning: format ‘%d’ expects a matching ‘int’ argument [-Wform
at=]
   12 | printf("Child Process: Type = %d, PID = %d, Parent PID = %d\n",getppid(
),getpid());
      |                                                           ~^
      |                                                            |
      |                                                            int
Q10Lab3.c:14:59: warning: format ‘%d’ expects a matching ‘int’ argument [-Wform
at=]
   14 | printf("Parent Process: Type = %d, PID = %d, Child PID = %d\n", getpid(
), child_pid);
      |                                                           ~^
      |                                                            |
      |                                                            int
treydenwilson@treydenwilson-VirtualBox:~$ ./Q10Lab3
Parent Process: Type = 1833, PID = 1834, Child PID = 1833
treydenwilson@treydenwilson-VirtualBox:~$ Child Process: Type = 808, PID = 1834
, Parent PID = -1756290869
```



```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/types.h>
4
5 int main() {
6
7 pid_t child_pid;
8
9 child_pid = fork();
10
11 if(child_pid == 0){
12 printf("Child Process: Type = %d, PID = %d, Parent PID =
   %d\n",getppid(),getpid());
13 }else{
14 printf("Parent Process: Type = %d, PID = %d, Child PID = %d\n", getpid(),
   child_pid);
15 }
16
17 return 0;
18
19 }
```