Operating Systems Laboratory (CS39002)
Assignment 3

Group 26
Seemant G. Achari (19CS10055) Rajas Bhatt (19CS30037)

Solution for Task 1 (b)

In general, there is a limit to the number of processes that can be created, so the `fork` call will fail if we demand the creation of a large number of child processes.
In our case, the `fork` call usually fails because of an `EAGAIN` error (Resource temporarily unavailable when `perror` is used), which is due to the system imposed limit on the number of threads. According to the man page of the `fork` call, this occurs due to one of the following situations:

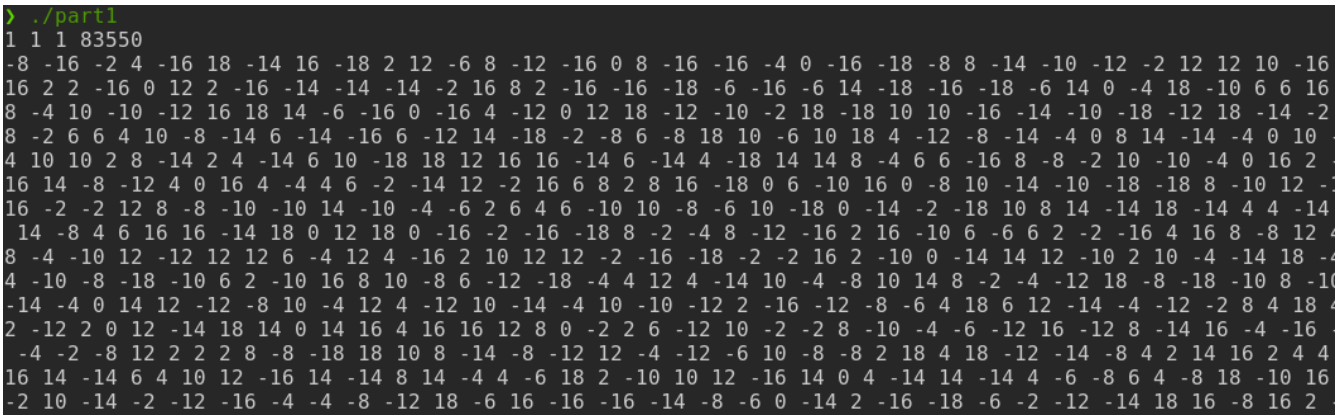
1. The limit on the number of processes. This is given by the `RLIMIT_NPROC` flag. This limit can be checked by typing in `ulimit -u`.
2. The system-wise limit on the number of processes and threads, given by `/proc/sys/kernel/threads-max`
3. The maximum number of PIDs was reached. This is given by `/proc/sys/kernel/pid_max`
4. The PID limit (`pids.max`) imposed by the process group was reached. This can be checked using `/sys/fs/cgroup/pids/user.slice/user-1000.slice/pids.max`

In our system, we find out that the values are as follows (in processes):

1. `ulimit -u`: 127628
2. `cat /proc/sys/kernel/threads-max: 255257`
3. `cat /proc/sys/kernel/pid_max: 4194304`
4. `cat /sys/fs/cgroup/pids/user.slice/user-1000.slice/pids.max: 84234`

It can be seen that the limiting value is given by the value of `pids.max` (84234). This is basically the limiting value for the maximum number of processes created in a group. Since fork assigns the same pgid to children, the maximum number of children that can be created are bounded by 84234.

We create r1 * c2 child processes in total, so the maximum value of r1 * c2 will be equal to this value (`pids.max`). In practice, after repeated attempts, it can be seen that the value of r1 * c2 less than 83600 works in practice. This may be due to some other process in the process group or due to some system constraints. Therefore, a value of r1 = 289 and c2 = 289 works (which makes r1 * c2 = 83521).



But r1 = 290 and c2 = 290 (which makes r1 * c2 = 84100) doesn't.