

Question #2

Part a

demo_uts_namespaces.c and ns_exec.c

```
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$ sudo su
[sudo] password for raja:
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# ps af
  PID TTY          STAT       TIME COMMAND
 27126 pts/4    Ss           0:00   /usr/bin/bash --init-file /usr/share/code/resources/app/out/vs/workbench/contrib/terminal/browser/media/shellIntegration-bash.sh
 27170 pts/4    S+           0:00   \_ sudo su
 27171 pts/5    Ss           0:00   \_ sudo su
 27172 pts/5    S            0:00   \_ su
 27173 pts/5    S            0:00   \_ bash
 27203 pts/5    R+           0:00   \_ ps af
 2138 tty2    Ssl+         0:00   /usr/libexec/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu
 2140 tty2    Sl+          13:20   \_ /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -nolisten tcp -background none -noreset -keeptty -novtswitch -verbose 3
 2197 tty2    Sl+          0:00   \_ /usr/libexec/gnome-session-binary --session=ubuntu
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# ./uts_namespaces raja
PID of child created by clone() is 27277 and parent PID is 27276
uts.nodename in child: raja
^Z
[1]+  Stopped                  ./uts_namespaces raja
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# jobs -l
[1]+  27276 Stopped                  ./uts_namespaces raja
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# readlink /proc/27276/ns/uts
uts:[4026531838]
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# readlink /proc/27277/ns/uts
uts:[4026532270]
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# ./ns_exec /proc/27277/ns/uts /bin/bash
root@raja:/home/raja/Desktop/Spring-2023/cs695/assignment-4# hostname
raja
root@raja:/home/raja/Desktop/Spring-2023/cs695/assignment-4# uname -n
raja
```

- `demo_uts_namespaces.c` is used to create a new child process with different uts namespaces using `clone`. The child process is created with `CLONE_NEWUTS` flag. The parent process waits for the child process to finish and then exits.
- I have recreated this example as asked in the question. Terminal output is shown in the above image.

Part b

```
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$ unshare -h
Usage:
unshare [options] [<program> [<argument>...]]

Run a program with some namespaces unshared from the parent.

Options:
-m, --mount[=<file>]      unshare mounts namespace
-u, --uts[=<file>]        unshare UTS namespace (hostname etc)
-i, --ipc[=<file>]        unshare System V IPC namespace
-n, --net[=<file>]        unshare network namespace
-p, --pid[=<file>]        unshare pid namespace
-U, --user[=<file>]        unshare user namespace
-C, --cgroup[=<file>]     unshare cgroup namespace
-T, --time[=<file>]       unshare time namespace

-f, --fork                fork before launching <program>
--map-user=<uid>|<name>    map current user to uid (implies --user)
--map-group=<gid>|<name>  map current group to gid (implies --user)
-r, --map-root-user       map current user to root (implies --user)
-c, --map-current-user    map current user to itself (implies --user)

--kill-child[=<signame>]  when dying, kill the forked child (implies --fork)
                        defaults to SIGKILL
--mount-proc[=<dir>]      mount proc filesystem first (implies --mount)
--propagation slave|shared|private|unchanged
                        modify mount propagation in mount namespace
--setgroups allow|deny   control the setgroups syscall in user namespaces
--keep-caps              retain capabilities granted in user namespaces

-R, --root=<dir>          run the command with root directory set to <dir>
-w, --wd=<dir>           change working directory to <dir>
-S, --setuid <uid>        set uid in entered namespace
-G, --setgid <gid>        set gid in entered namespace
--monotonic <offset>     set clock monotonic offset (seconds) in time namespaces
--boottime <offset>      set clock boottime offset (seconds) in time namespaces

-h, --help                display this help
-V, --version              display version

For more details see unshare(1).
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$

raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$ nsenter -h
Usage:
nsenter [options] [<program> [<argument>...]]

Run a program with namespaces of other processes.

Options:
-a, --all                enter all namespaces
-t, --target <pid>       target process to get namespaces from
-m, --mount[=<file>]     enter mount namespace
-u, --uts[=<file>]        enter UTS namespace (hostname etc)
-i, --ipc[=<file>]        enter System V IPC namespace
-n, --net[=<file>]        enter network namespace
-p, --pid[=<file>]        enter pid namespace
-C, --cgroup[=<file>]     enter cgroup namespace
-U, --user[=<file>]       enter user namespace
-T, --time[=<file>]       enter time namespace
-S, --setuid <uid>        set uid in entered namespace
-G, --setgid <gid>        set gid in entered namespace
--preserve-credentials  do not touch uids or gids
-r, --root[=<dir>]        set the root directory
-w, --wd[=<dir>]          set the working directory
-F, --no-fork             do not fork before exec'ing <program>
-Z, --follow-context      set SELinux context according to --target PID

-h, --help                display this help
-V, --version              display version

For more details see nsenter(1).
raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$
```

```

raja@ubuntu22:~$ ps af
  PID TTY          STAT       TIME COMMAND
 31127 pts/9        Ss+        0:00 bash
 31142 pts/9        S+         0:00 \_ sudo su
 31144 pts/10       Ss         0:00 \_ sudo su
 31145 pts/10       S          0:00 \_ su
 31146 pts/10       S+         0:00 \_ bash
 31107 pts/8        Ss         0:00 bash
 31358 pts/8        R+         0:00 \_ ps af
 31028 pts/7        Ss+        0:00 /usr/bin/bash --init-file /usr/share/code/resources/app/out/vs/workbench
 2138 tty2         Ssl+       0:00 /usr/libexec/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubu
 2140 tty2         Sl+        15:10 \_ /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthor
 2197 tty2         Sl+        0:00 \_ /usr/libexec/gnome-session-binary --session=ubuntu
raja@ubuntu22:~$ ps af
  PID TTY          STAT       TIME COMMAND
 31127 pts/9        Ss         0:00 bash
 31142 pts/9        S+         0:00 \_ sudo su
 31144 pts/10       Ss         0:00 \_ sudo su
 31145 pts/10       S          0:00 \_ su
 31146 pts/10       S          0:00 \_ bash
 31354 pts/10       S          0:00 \_ unshare --uts --pid --fork -r --mount-proc
 31365 pts/10       S+         0:00 \_ -bash
 31107 pts/8        Ss         0:00 bash
 31387 pts/8        R+         0:00 \_ ps af
 31028 pts/7        Ss+        0:00 /usr/bin/bash --init-file /usr/share/code/resources/app/out/vs/workbench
 2138 tty2         Ssl+       0:00 /usr/libexec/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubu
 2140 tty2         Sl+        15:14 \_ /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthor
 2197 tty2         Sl+        0:00 \_ /usr/libexec/gnome-session-binary --session=ubuntu
raja@ubuntu22:~$ sudo nsenter --target 31365 -a
root@raja:/# hostname
raja
root@raja:/# ps af
  PID TTY          STAT       TIME COMMAND
   18 pts/11       S          0:00 -bash
   33 pts/11       R+         0:00 \_ ps af
    1 pts/10       S+         0:00 -bash
root@raja:/# Killed
raja@ubuntu22:~$

root@ubuntu22:/home/raja# hostname
ubuntu22
root@ubuntu22:/home/raja# unshare --uts --pid --fork -r --mount-proc
root@ubuntu22:/home/raja# hostname
raja
root@ubuntu22:/home/raja# ps af
  PID TTY          STAT       TIME COMMAND
    1 pts/10       S          0:00 -bash
   17 pts/10       R+         0:00 ps af
root@ubuntu22:/home/raja# ps af
  PID TTY          STAT       TIME COMMAND
   18 pts/11       S+         0:00 -bash
   34 pts/10       R+         0:00 ps af
root@ubuntu22:/home/raja# exit
logout
root@ubuntu22:/home/raja#

```

- This part is similar to the part a. The only difference is that we are replicating the behavior of part a using `unshare` and `nsenter` command line tools. Terminal output is shown in the above image.

part c

```

o raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$ sudo su
[sudo] password for raja:
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# hostname
ubuntu22
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# ./uts_namespaces raja
PID of child created by clone() is 38366 and parent PID is 38365
uts.nodename in child: raja
uts.nodename in parent: ubuntu22
o raja@ubuntu22:~/Desktop/Spring-2023/cs695/assignment-4$ sudo su
[sudo] password for raja:
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# hostname
ubuntu22
root@ubuntu22:/home/raja/Desktop/Spring-2023/cs695/assignment-4# ./ns_pid 38366 /bin/b
ash
root@raja:/home/raja/Desktop/Spring-2023/cs695/assignment-4# hostname
raja

```

- `ns_pid.c` takes a process id as an argument and then join the namespace of the process. It then execute the program passed as an argument in the namespace of the process.
- `ns_pid.c` can be found in the `part3` directory.
- Terminal output is shown in the above image.