Operation Systems , 2023
HOMEWORK 3

系級: 資管碩一　　學號: 112356045　　姓名: 謝東睿

1. **說明 clone() 與 fork() 的差異：**

   fork( )不需要任何參數，僅僅是在創建一個child process，並為其創建一個獨立於parant process的空間。fork()在創建一個進程時，child process完全複製parent process的資源，複製出來的child process有自己的 task_struct 結構和 pid，但卻複製parent process的其它所有資源。

   clone( ) 帶有參數 ， 可以指定創建新的命名空間（namespace），還可以將parent process的資源有選擇地複製給child process，而沒有複製的資料結構則通過pointer的複製讓child process共享，具體要複製哪些資源給child process，則由參數列表中的clone_flags決定。

2. **UTS namespace 截圖 x3：**

```
root@m1maclinun:/home/dongrui/os_hw3# hostname
m1maclinun
root@11fe2f3062bf:/# hostname
11fe2f3062bf
dongrui@m1maclinunx:~/os_hw3$ hostname
m1maclinunx
```

3. **IPC namespace 截圖 x3：**

```
root@m1maclinun:/home/dongrui/os_hw3# ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
0xcf7f6888 2          root       644        0            0

root@0359300488ec:/# ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
0x1267e721 0          root       644        0            0

root@m1maclinunx:/home/dongrui/os_hw3# ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
0x7b46b9a9 0          root       644        0            0
```

4. PID namespace 截圖 x3：

```
root@m1maclinun:/home/dongrui/os_hw3# ls -l /proc/$$/ns
total 0
lrwxrwxrwx 1 root root 0 Jan  5 06:55 cgroup -> 'cgroup:[4026531835]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 ipc -> 'ipc:[4026531839]'
lrwxrwxrwx 1 root root 0 Jan  5 06:55 mnt -> 'mnt:[4026531841]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 net -> 'net:[4026531840]'
lrwxrwxrwx 1 root root 0 Jan  5 06:58 pid -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 pid_for_children -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 time -> 'time:[4026531834]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 time_for_children -> 'time:[4026531834]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 user -> 'user:[4026531837]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 uts -> 'uts:[4026531838]'
root@0359300488ec:/# ls -l /proc/$$/ns
total 0
lrwxrwxrwx 1 root root 0 Jan  5 08:15 cgroup -> 'cgroup:[4026532488]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 ipc -> 'ipc:[4026532425]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 mnt -> 'mnt:[4026532423]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 net -> 'net:[4026532427]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 pid -> 'pid:[4026532426]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 pid_for_children -> 'pid:[4026532426]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 time -> 'time:[4026531834]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 time_for_children -> 'time:[4026531834]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 user -> 'user:[4026531837]'
lrwxrwxrwx 1 root root 0 Jan  5 08:15 uts -> 'uts:[4026532424]'
root@m1maclinunx:/home/dongrui/os_hw3# ls -l /proc/$$/ns
total 0
lrwxrwxrwx 1 root root 0 Jan  5 06:55 cgroup -> 'cgroup:[4026531835]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 ipc -> 'ipc:[4026531839]'
lrwxrwxrwx 1 root root 0 Jan  5 06:55 mnt -> 'mnt:[4026531841]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 net -> 'net:[4026531840]'
lrwxrwxrwx 1 root root 0 Jan  5 06:58 pid -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 pid_for_children -> 'pid:[4026531836]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 time -> 'time:[4026531834]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 time_for_children -> 'time:[4026531834]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 user -> 'user:[4026531837]'
lrwxrwxrwx 1 root root 0 Jan  5 08:14 uts -> 'uts:[4026531838]'
```

5. MNT namespace 截圖 x3：

```
root@m1maclinun:/home/dongrui/os_hw3# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2 167924 11368 ?        Ss   06:55   0:03 /sbin/init
root           2  0.0  0.0      0     0 ?        S    06:55   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   06:55   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   06:55   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   06:55   0:00 [slub_flushwq
root           6  0.0  0.0      0     0 ?        I<   06:55   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   06:55   0:00 [kworker/0:0H
root          10  0.0  0.0      0     0 ?        I<   06:55   0:00 [mm_percpu_wq
root          11  0.0  0.0      0     0 ?        S    06:55   0:00 [rcu_tasks_ru
root          12  0.0  0.0      0     0 ?        S    06:55   0:00 [rcu_tasks_tr
root          13  0.0  0.0      0     0 ?        S    06:55   0:00 [ksoftirqd/0]
root          14  0.0  0.0      0     0 ?        I    06:55   0:00 [rcu_sched]
root          15  0.0  0.0      0     0 ?        S    06:55   0:00 [migration/0]
root          16  0.0  0.0      0     0 ?        S    06:55   0:00 [idle_inject/
root          17  0.0  0.0      0     0 ?        I    06:55   0:02 [kworker/0:1-
root          18  0.0  0.0      0     0 ?        S    06:55   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    06:55   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    06:55   0:00 [idle_inject/
root          21  0.0  0.0      0     0 ?        S    06:55   0:00 [migration/1]
root          22  0.0  0.0      0     0 ?        S    06:55   0:00 [ksoftirqd/1]
root          24  0.0  0.0      0     0 ?        I<   06:55   0:00 [kworker/1:0H
root          25  0.0  0.0      0     0 ?        S    06:55   0:00 [cpuhp/2]
root          26  0.0  0.0      0     0 ?        S    06:55   0:00 [idle_inject/
root          27  0.0  0.0      0     0 ?        S    06:55   0:00 [migration/2]
root          28  0.0  0.0      0     0 ?        S    06:55   0:00 [ksoftirqd/2]
root          30  0.0  0.0      0     0 ?        I<   06:55   0:00 [kworker/2:0H
root          31  0.0  0.0      0     0 ?        S    06:55   0:00 [cpuhp/3]
root          32  0.0  0.0      0     0 ?        S    06:55   0:00 [idle_inject/
root          33  0.0  0.0      0     0 ?        S    06:55   0:00 [migration/3]
root          34  0.0  0.0      0     0 ?        S    06:55   0:00 [ksoftirqd/3]
root          35  0.0  0.0      0     0 ?        I    06:55   0:03 [kworker/3:0-
root          36  0.0  0.0      0     0 ?        I<   06:55   0:00 [kworker/3:0H
root          37  0.0  0.0      0     0 ?        S    06:55   0:00 [kdevtmpfs]
root          38  0.0  0.0      0     0 ?        I<   06:55   0:00 [inet_frag_wq
root          40  0.0  0.0      0     0 ?        I    06:55   0:01 [kworker/1:1-
root          42  0.0  0.0      0     0 ?        S    06:55   0:00 [kauditd]
root          43  0.0  0.0      0     0 ?        S    06:55   0:00 [khungtaskd]
root          44  0.0  0.0      0     0 ?        S    06:55   0:00 [oom_reaper]
root          45  0.0  0.0      0     0 ?        I<   06:55   0:00 [writeback]
root          46  0.0  0.0      0     0 ?        S    06:55   0:00 [kcompactd0]
root          47  0.0  0.0      0     0 ?        SN   06:55   0:00 [ksmd]
root          48  0.0  0.0      0     0 ?        SN   06:55   0:00 [khugepaged]
root          94  0.0  0.0      0     0 ?        I<   06:55   0:00 [kintegrityd]
root          95  0.0  0.0      0     0 ?        I<   06:55   0:00 [kblockd]
root          96  0.0  0.0      0     0 ?        I<   06:55   0:00 [blkcg_punt_b
root          97  0.0  0.0      0     0 ?        I<   06:55   0:00 [tpm_dev_wq]
root          98  0.0  0.0      0     0 ?        I<   06:55   0:00 [ata_sff]
root          99  0.0  0.0      0     0 ?        I<   06:55   0:00 [md]
root         100  0.0  0.0      0     0 ?        I<   06:55   0:00 [edac-poller]
root         101  0.0  0.0      0     0 ?        I<   06:55   0:00 [devfreq_wq]
root         102  0.0  0.0      0     0 ?        S    06:55   0:00 [watchdogd]
root         104  0.0  0.0      0     0 ?        I<   06:55   0:00 [kworker/1:1H
root         105  0.0  0.0      0     0 ?        S    06:55   0:00 [kswapd0]
root         106  0.0  0.0      0     0 ?        S    06:55   0:00 [ecryptfs-kth
root         108  0.0  0.0      0     0 ?        I<   06:55   0:00 [kthrotld]
root         109  0.0  0.0      0     0 ?        S    06:55   0:00 [irq/49-ACPI:
root         110  0.0  0.0      0     0 ?        I<   06:55   0:00 [acpi_thermal
root         112  0.0  0.0      0     0 ?        I<   06:55   0:00 [mld]
root         113  0.0  0.0      0     0 ?        I<   06:55   0:00 [ipv6_addrcon

root@0359300488ec:/# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   4136  3408 pts/0    Ss   08:10   0:00 /bin/bash
root          18  0.0  0.0   6412  1648 pts/0    R+   08:28   0:00 ps aux
```

```
dongrui     1534  0.0  0.1 106300  5652 ?        S    06:55   0:00 (sd-pam)
dongrui     1541  0.0  0.1   8256  5088 tty1     S    06:55   0:00 -bash
root        1808  0.0  0.4 297552 18840 ?        Ssl  06:57   0:00 /usr/libexec/packagekitd
root        2234  0.2  1.0 1344572 43520 ?       Ssl  06:58   0:14 /usr/bin/containerd
root        2532  0.1  1.8 1591800 72112 ?       Ssl  06:58   0:05 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
root        2877  0.0  0.0      0     0 ?        I    07:00   0:00 [kworker/2:0-events]
root        3532  0.0  0.0      0     0 ?        I    07:10   0:00 [kworker/3:1]
root        3545  0.0  0.2  18400  9776 ?        Ss   07:28   0:00 sshd: dongrui [priv]
dongrui     3600  0.0  0.1  18532  6532 ?        S    07:28   0:00 sshd: dongrui@pts/0
dongrui     3601  0.0  0.1   8244  5060 pts/0    Ss   07:28   0:00 -bash
root        3623  0.0  0.0      0     0 ?        I    07:42   0:00 [kworker/u8:0-events_power_efficient]
root        3639  0.0  0.1  13020  4788 pts/0    S+   07:46   0:00 sudo ./uts
root        3640  0.0  0.0  13020  1840 pts/1    Ss   07:46   0:00 sudo ./uts
root        3641  0.0  0.0   3212   764 pts/1    S    07:46   0:00 ./uts
root        3642  0.0  0.0   7144  3836 pts/1    S    07:46   0:00 /bin/bash
root        3812  0.0  0.0      0     0 ?        I    07:55   0:01 [kworker/u8:3-events_unbound]
root        3833  0.0  0.1  13400  4780 pts/1    S+   08:00   0:00 sudo ./ipc
root        3834  0.0  0.0  13400  1836 pts/4    Ss   08:00   0:00 sudo ./ipc
root        3835  0.0  0.0   3212   752 pts/4    S    08:00   0:00 ./ipc
root        3836  0.0  0.0   7144  3840 pts/4    S    08:00   0:00 /bin/bash
root        3878  0.0  0.1  12988  4788 tty1     R+   08:05   0:00 sudo ./ipc
root        3879  0.0  0.0  12988   676 pts/5    Ss   08:05   0:00 sudo ./ipc
root        3880  0.0  0.0   3212   760 pts/5    S    08:05   0:00 ./ipc
root        3881  0.0  0.0   7144  3860 pts/5    S    08:05   0:00 /bin/bash
root        3903  0.0  0.0      0     0 ?        R    08:08   0:00 [kworker/u8:1-events_unbound]
root        3910  0.0  0.0      0     0 ?        I    08:09   0:00 [kworker/2:1-kdmflush]
root        3936  0.0  0.0      0     0 ?        R    08:09   0:00 [kworker/1:0+events]
root        3952  0.0  0.2  18400  9648 ?        Ss   08:09   0:00 sshd: dongrui [priv]
dongrui     4048  0.0  0.1  18532  6524 ?        S    08:09   0:00 sshd: dongrui@pts/2
dongrui     4049  0.0  0.1   8244  5068 pts/2    Ss   08:09   0:00 -bash
root        4058  0.0  0.1  13020  4784 pts/2    S+   08:10   0:00 sudo docker run -it ubuntu /bin/bash
root        4059  0.0  0.0  13020  1836 pts/3    Ss   08:10   0:00 sudo docker run -it ubuntu /bin/bash
root        4060  0.0  0.5 1401236 23288 pts/3   Sl+  08:10   0:00 docker run -it ubuntu /bin/bash
root        4100  0.0  0.3 721968 12720 ?        Sl   08:10   0:00 /usr/bin/containerd-shim-runc-v2 -namespace moby -id 0359300488ec0f870a38235705a487d1781cc701
root        4120  0.0  0.0   4136  3408 pts/0    Ss+  08:10   0:00 /bin/bash
root        4159  0.0  0.1  12984  4776 pts/4    S+   08:14   0:00 sudo ./pid
root        4160  0.0  0.0  12984   676 pts/6    Ss   08:14   0:00 sudo ./pid
root        4161  0.0  0.0   3212   768 pts/6    S    08:14   0:00 ./pid
root        4162  0.0  0.0   7144  3852 pts/6    S+   08:14   0:00 /bin/bash
root        4174  0.0  0.0      0     0 ?        I    08:16   0:00 [kworker/0:0-events]
root        4175  0.0  0.0   5592   808 tty2     Ss+  08:16   0:00 /sbin/agetty -o -p -- \u --noclear tty2 linux
root        4184  0.0  0.1  13400  4788 pts/5    S+   08:17   0:00 sudo ./pid
root        4185  0.0  0.0  13400  1832 pts/7    Ss   08:17   0:00 sudo ./pid
root        4186  0.0  0.0   3212   752 pts/7    S    08:17   0:00 ./pid
root        4187  0.0  0.0   7144  3856 pts/7    S    08:17   0:00 /bin/bash
root        4201  0.1  0.0      0     0 ?        I    08:23   0:00 [kworker/u8:2-ext4-rsv-conversion]
root        4210  0.0  0.0      0     0 ?        I    08:24   0:00 [kworker/0:2-events]
root        4246  0.0  0.0      0     0 ?        I    08:26   0:00 [kworker/2:2]
root        4253  0.0  0.0   9420  1652 pts/7    R+   08:26   0:00 ps aux
```

6. Docker 上使用資源截圖 截圖 x2：

```
CONTAINER ID   NAME             CPU %     MEM USAGE / LIMIT     MEM %     NET I/O
       BLOCK I/O   PIDS
5020fe3ba23c   stress           90.64%    5.742MiB / 3.817GiB   0.15%     726B /
0B     0B / 0B     2
18d64b321a5a   musing_rhodes    0.00%     820KiB / 3.817GiB     0.02%     1.46kB
/ 0B   0B / 0B     1
```

```
CONTAINER ID   NAME             CPU %     MEM USAGE / LIMIT     MEM %     NET I/O
       BLOCK I/O   PIDS
6ed4b9f80fa1   stress           20.88%    5.738MiB / 3.817GiB   0.15%     796B /
0B     0B / 0B     2
18d64b321a5a   musing_rhodes    0.00%     820KiB / 3.817GiB     0.02%     1.53kB
/ 0B   0B / 0B     1
```

7. 使用 Cgroup 操做 deadloop.c 的資源使用 截圖 x2：

```
top - 10:14:15 up 33 min,  4 users,  load average: 0.69, 0.71, 0.63
Tasks: 110 total,   2 running, 108 sleeping,   0 stopped,   0 zombie
%Cpu(s): 99.3 us,  0.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3909.0 total,   2205.0 free,    274.5 used,   1429.5 buff/cache
MiB Swap:   3908.0 total,   3908.0 free,      0.0 used.   3375.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3846 root      20   0    2640    956    868 R  98.0   0.0   0:46.37 deadloop
   3887 dongrui   20   0   10472   3692   3136 R   0.7   0.1   0:00.21 top
   3889 root      20   0       0      0      0 I   0.7   0.0   0:00.02 kworker+
   2484 root      20   0 1211232  49048  34744 S   0.3   1.2   0:06.52 contain+
   3231 dongrui   20   0   17300   8016   5608 S   0.3   0.2   0:00.53 sshd
   3261 root      20   0       0      0      0 I   0.3   0.0   0:01.62 kworker+
      1 root      20   0  167676  13004   8156 S   0.0   0.3   0:33.44 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
      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_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      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   0:06.11 kworker+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
top - 10:21:05 up 40 min,  6 users,  load average: 0.60, 0.81, 0.73
Tasks: 113 total,   2 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s): 20.7 us,  1.3 sy,  0.0 ni, 78.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3909.0 total,   2203.8 free,    275.7 used,   1429.6 buff/cache
MiB Swap:   3908.0 total,   3908.0 free,      0.0 used.   3374.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3846 root      20   0    2640    956    868 R  19.7   0.0   6:15.26 deadloop
   3982 dongrui   20   0   10472   3728   3172 R   1.6   0.1   0:00.58 top
   3261 root      20   0       0      0      0 I   0.7   0.0   0:03.34 kworker+
     25 root      20   0       0      0      0 S   0.3   0.0   0:00.91 kcompac+
   2484 root      20   0 1211232  48932  34808 S   0.3   1.2   0:07.61 contain+
   3231 dongrui   20   0   17300   8016   5608 S   0.3   0.2   0:01.27 sshd
      1 root      20   0  167676  13004   8156 S   0.0   0.3   0:33.53 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
      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_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   0:02.31 ksoftir+
```

8. **Runc - 使用 ps aux 指令列出容器下的 process 截圖 x1：**

```
dongrui@dongrui:~/os_hw3$ ps aux | grep 4457
root        4457  0.0  0.0   1632     4 pts/0    Ss+  10:39   0:00 sh
dongrui     4571  0.0  0.0   6476  2148 pts/4    S+   10:46   0:00 grep --color=
auto 4457
```

9. **心得:**

(請寫下完成本次作業的心得、學到哪些東西、困難點的部分。)

我覺得這次的作業雖然有很多道步驟需要操作，但是基本上內容主要都是對比在 VM terminal、Docker container、src 資料夾裡 .c 檔的差異，或是觀察限制 CPU後的使用率為何。從這次作業中，我可以發現到原來使用 namespace 能做到隔離的動

作，讓不同的終端機介面可以產生不同的 output，其中每個不一樣的 namespace（uts、ipc、pid、mnt）都有著各自的功用，對於開發者來說可以依照自身的需求去做搭配使用，感覺會是相當方便的使用手法。接著，對於 Docker 上使用資源限制與Cgroup的使用方法，我覺得可以限制CPU的使用率這一點算是很好的功能，因為有時候可能會希望某項東西不要佔用 CPU 太高的使用率，像這種時候就能使用從這次作業中學到的方法來限制CPU的使用率，非常好用。