**Operation Systems , 2023**

**HOMEWORK 3**

系級: 資管碩一 學號: 112356045 姓名: 謝東睿

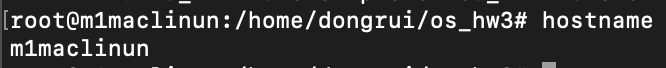
1. **說明clone() 與 fork() 的差異：**

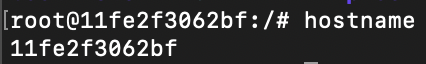
fork( )不需要任何參數，僅僅是在創建一個child process，並為

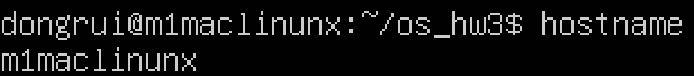
其創建一個獨立於parant process的空間。fork()在創建一個進程時，child process完全複製parent process的資源，複製出來的child process有自己的 task\_struct 結構和 pid，但卻複製parent process的其它所有資源。

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。

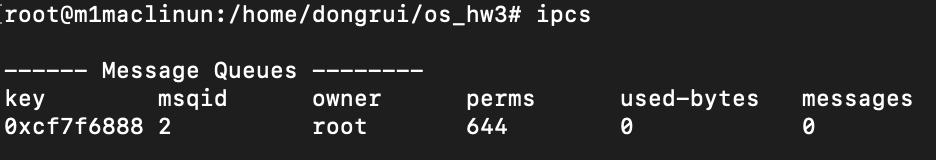
1. **UTS namespace 截圖x3：**

****

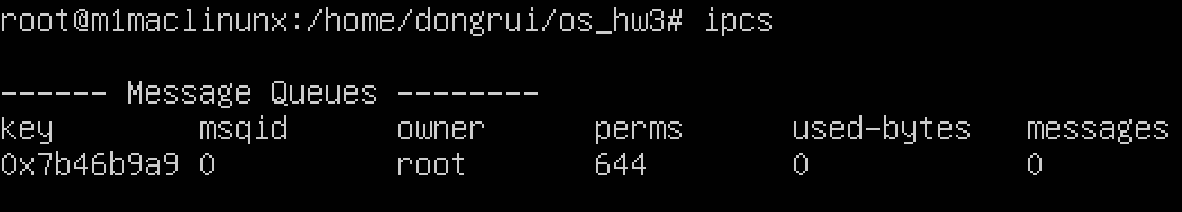
****

****

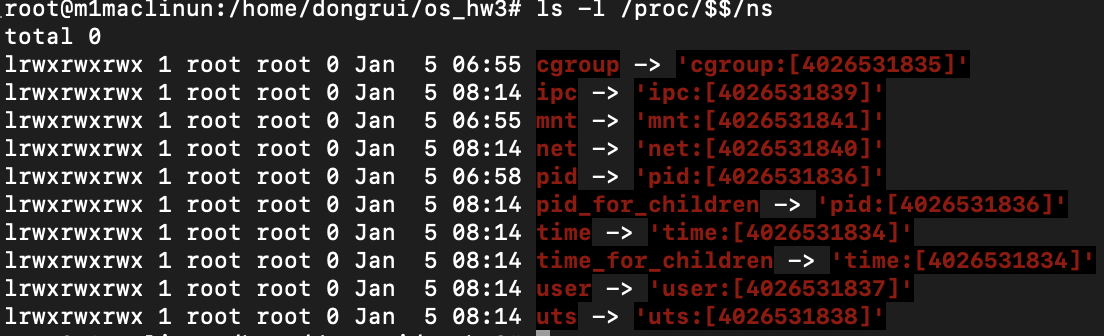
1. **IPC namespace 截圖x3：**

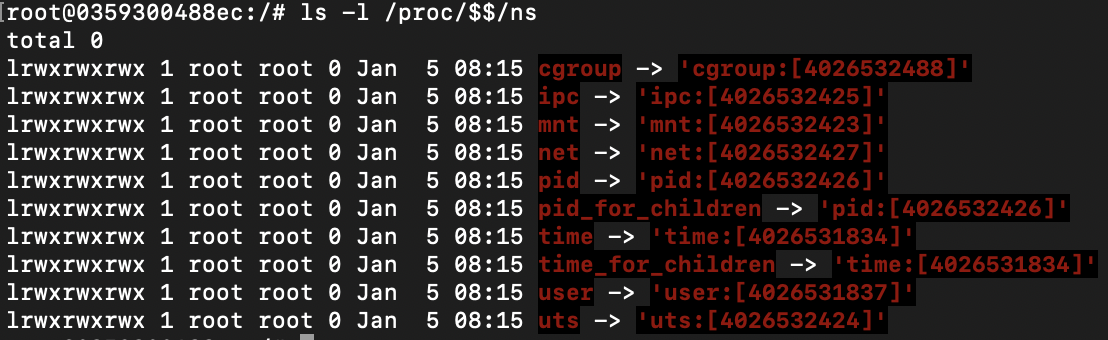
****

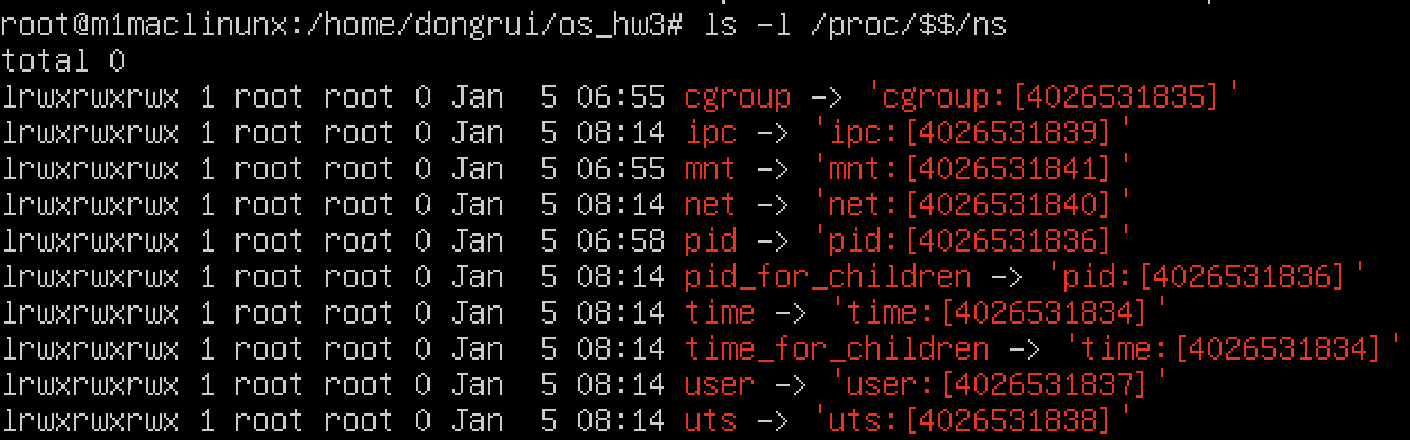
****

****

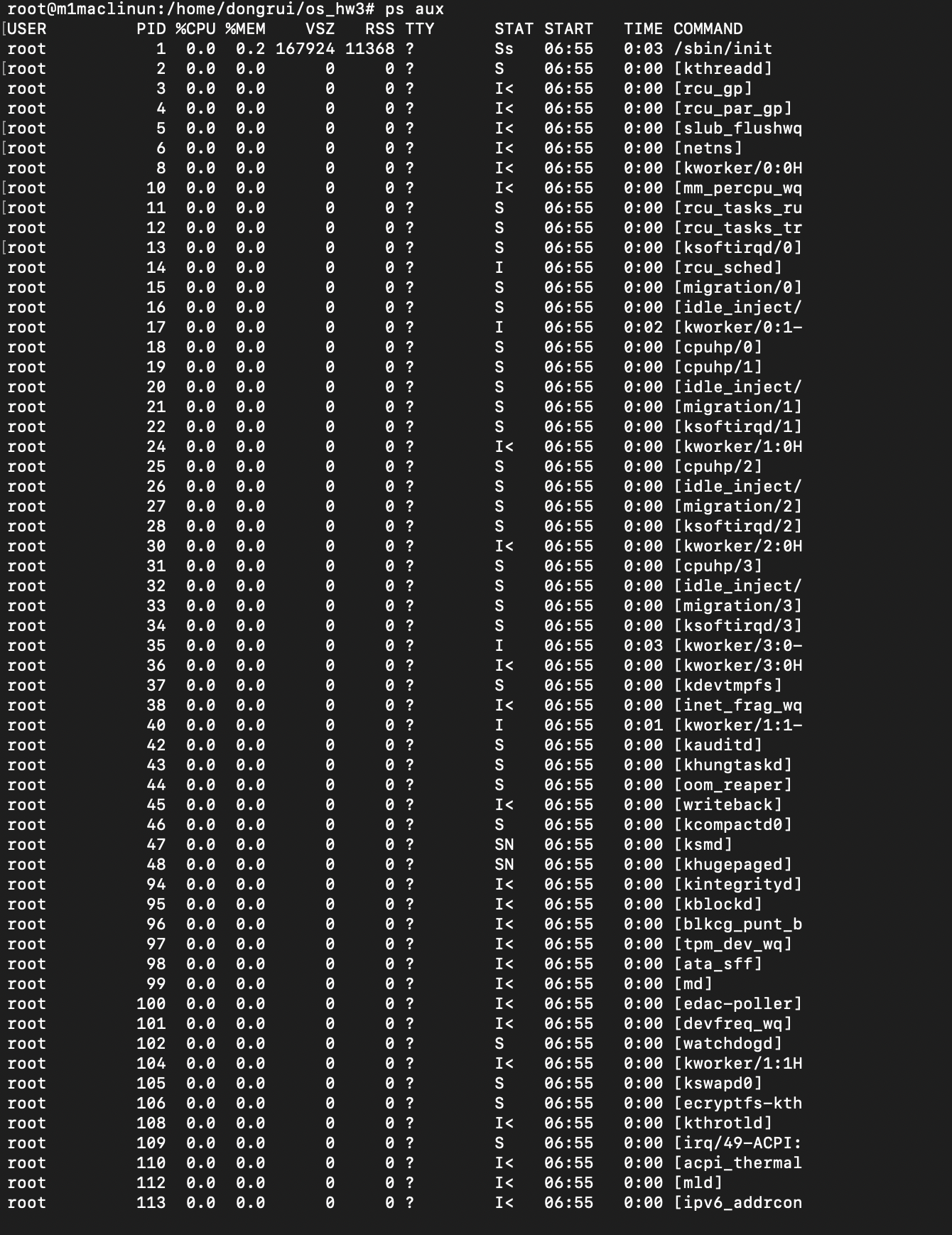
1. **PID namespace 截圖x3：**

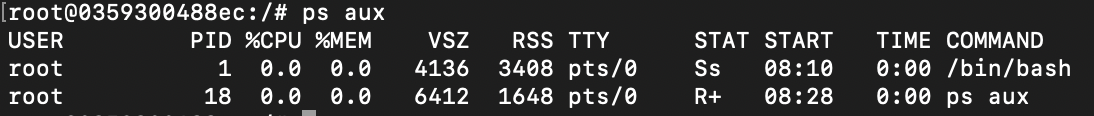
****

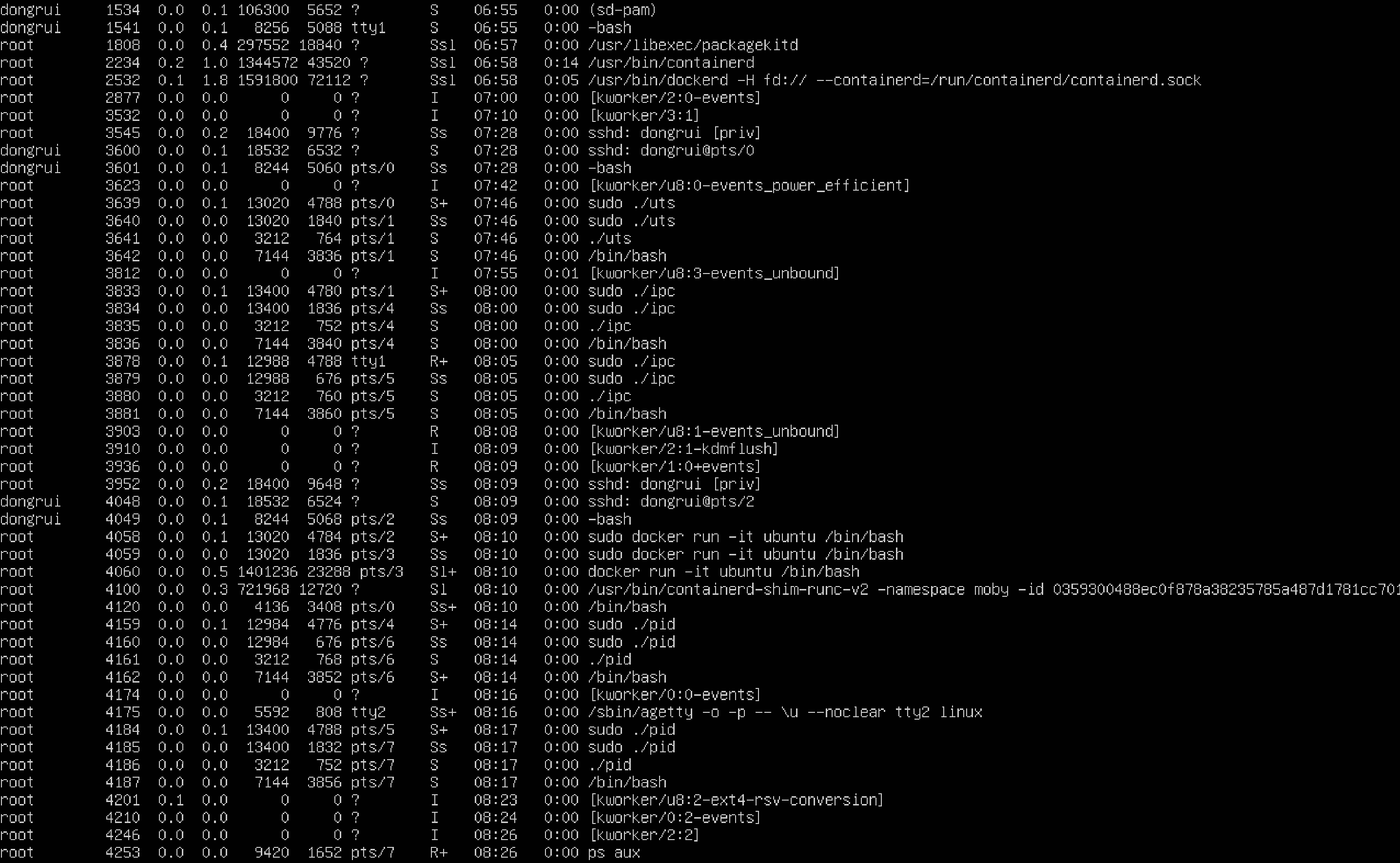
****

****

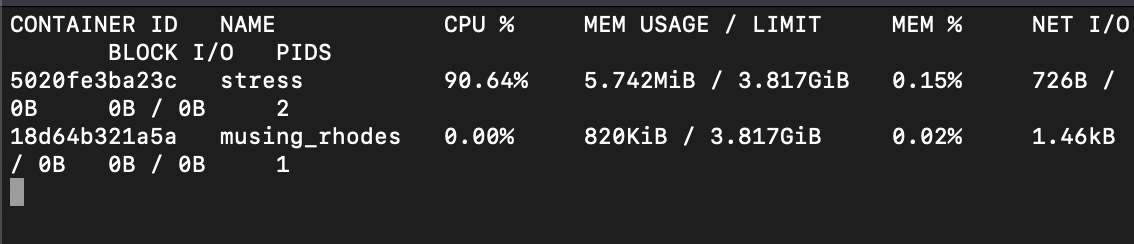
1. **MNT namespace 截圖x3：**

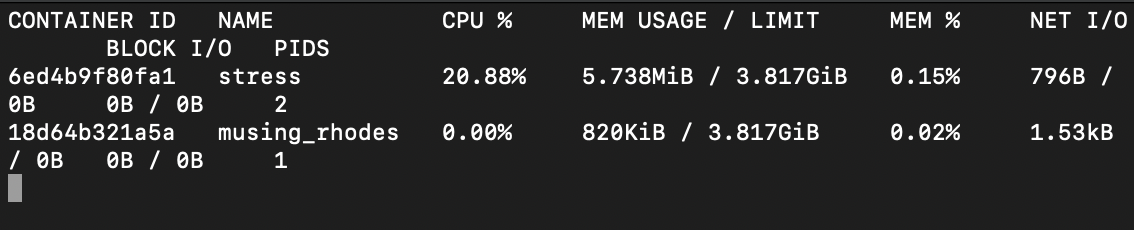
****

****

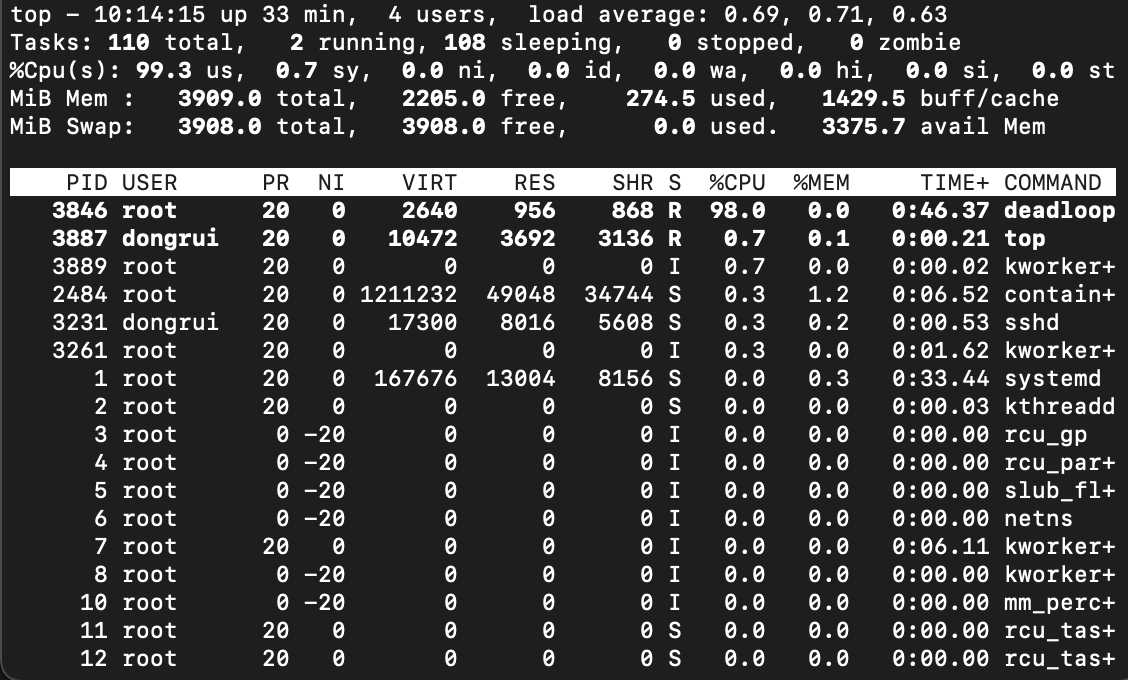
****

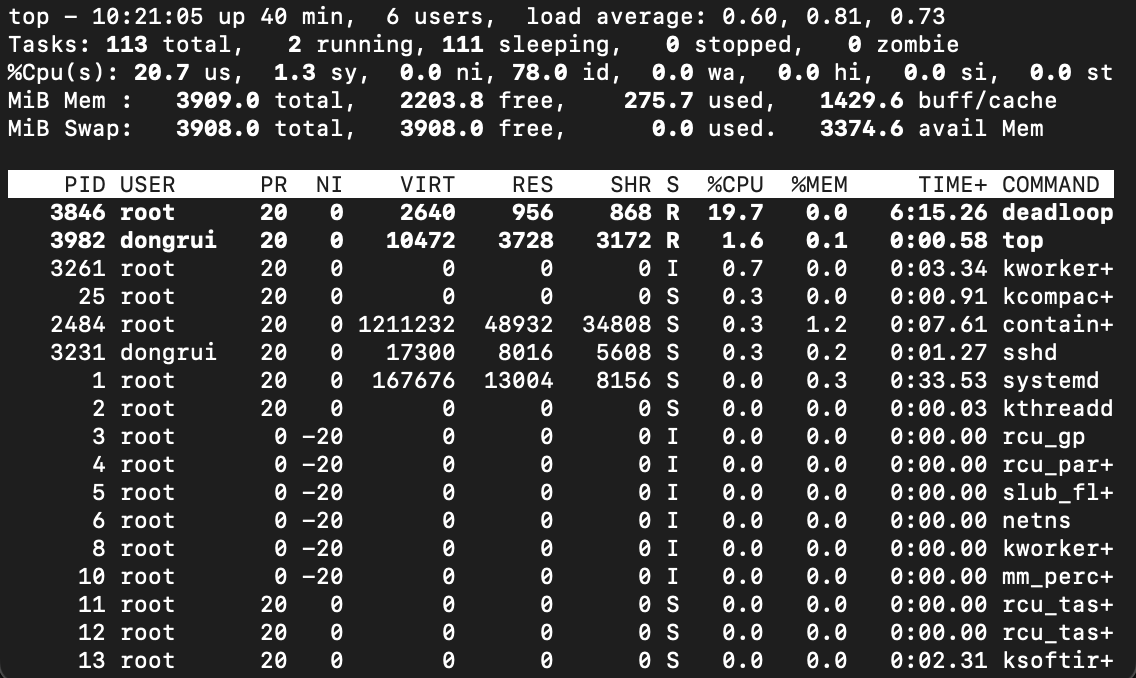
1. **Docker 上使用資源截圖 截圖x2：**

****

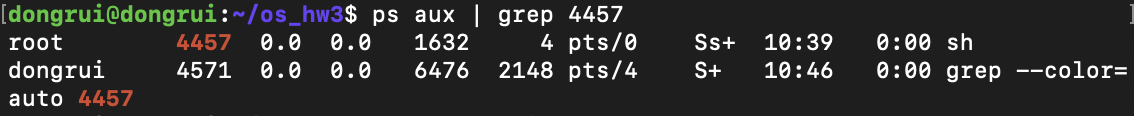
****

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





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



1. **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。