Exercise 2.
Answer Sheet

Student's Name: Tomonori Masubuchi Student's ID: s1240078

Problem 1. **Following instructions on the OS course WWW-site, load the Linux virtual machine into your browser and create the executable file loop by compiling the file loop.c using the following command for C compiler**

```
tcc loop.c -o loop
```

Answer to the following questions.

1) Using command "top", check and show the status of all processes on your virtual machine launched. Checking should be performed every second. "Sleep" means waiting.

Put your answer here.

```
Mem: 5632K used, 116816K free, 8K shrd, 0K buff, 712K cached
CPU:   4% usr   1% sys   0% nic  93% idle   0% io   0% irq   0% sirq
Load average: 0.00 0.00 0.00 1/30 91
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
    5     2 root     SW       0   0%   0% [kworker/u2:0]
    3     2 root     SW       0   0%   0% [kworker/0:0]
    8     2 root     SW       0   0%   0% [kdevtmpfs]
    9     2 root     SW       0   0%   0% [oom_reaper]
    6     2 root     SW<      0   0%   0% [mm_percpu_wq]
   11     2 root     SW       0   0%   0% [kcompactd0]
   12     2 root     SW<      0   0%   0% [crypto]
   13     2 root     SW<      0   0%   0% [bioset]
   10     2 root     SW<      0   0%   0% [writeback]
    2     0 root     SW       0   0%   0% [kthreadd]
   16     2 root     SW       0   0%   0% [kswapd0]
    4     2 root     SW<      0   0%   0% [kworker/0:0H]
   34     2 root     SW       0   0%   0% [khvcd]
   35     2 root     SW<      0   0%   0% [bioset]
   36     2 root     SW<      0   0%   0% [bioset]
   37     2 root     SW<      0   0%   0% [bioset]
   38     2 root     SW<      0   0%   0% [bioset]
   39     2 root     SW<      0   0%   0% [bioset]
   40     2 root     SW<      0   0%   0% [bioset]
   41     2 root     SW<      0   0%   0% [bioset]
   42     2 root     SW<      0   0%   0% [bioset]
   14     2 root     SW<      0   0%   0% [kblockd]
   15     2 root     SW       0   0%   0% [kworker/0:1]
   70     2 root     SW       0   0%   0% [kworker/u2:1]
   17     2 root     SW<      0   0%   0% [bioset]
```

2 Point a difference in system behavior if program **loop** will be launched by two following commands

a). **./loop**

b). **./loop&**

After which command the user can watch states of processes using the command **top**

Put your answer here.
We can operate in duplicate by using ‘&’.

Problem 2. Answer the following questions.

a). Use the "top" command and check the execution status of program **loop.c** every two seconds.

Put your answer here.

Command: top –d 2

```
Mem: 5780K used, 116668K free, 4K shrd, 0K buff, 712K cached
CPU:  98% usr   1% sys   0% nic   0% idle   0% io   0% irq   0% sirq
Load average: 1.13 1.02 0.57 2/31 110
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
  101   100 root     R     1112   1%   5% top -d 2
  100     1 root     S     1116   1%   0% sh -l
    1     0 root     S     1112   1%   0% {init} /bin/sh /sbin/init
   52     1 root     S      944   1%   0% dhcpcd
   15     2 root     SW       0   0%   0% [kworker/0:1]
    7     2 root     SW       0   0%   0% [ksoftirqd/0]
    8     2 root     SW       0   0%   0% [kdevtmpfs]
    9     2 root     SW       0   0%   0% [oom_reaper]
    6     2 root     SW<      0   0%   0% [mm_percpu_wq]
   11     2 root     SW       0   0%   0% [kcompactd0]
   12     2 root     SW<      0   0%   0% [crypto]
   13     2 root     SW<      0   0%   0% [bioset]
   10     2 root     SW<      0   0%   0% [writeback]
    2     0 root     SW       0   0%   0% [kthreadd]
    3     2 root     SW       0   0%   0% [kworker/0:0]
    4     2 root     SW<      0   0%   0% [kworker/0:0H]
    5     2 root     SW       0   0%   0% [kworker/u2:0]
   35     2 root     SW<      0   0%   0% [bioset]
   36     2 root     SW<      0   0%   0% [bioset]
   37     2 root     SW<      0   0%   0% [bioset]
   38     2 root     SW<      0   0%   0% [bioset]
   39     2 root     SW<      0   0%   0% [bioset]
   40     2 root     SW<      0   0%   0% [bioset]
   41     2 root     SW<      0   0%   0% [bioset]
   42     2 root     SW<      0   0%   0% [bioset]
   14     2 root     SW<      0   0%   0% [kblockd]
```

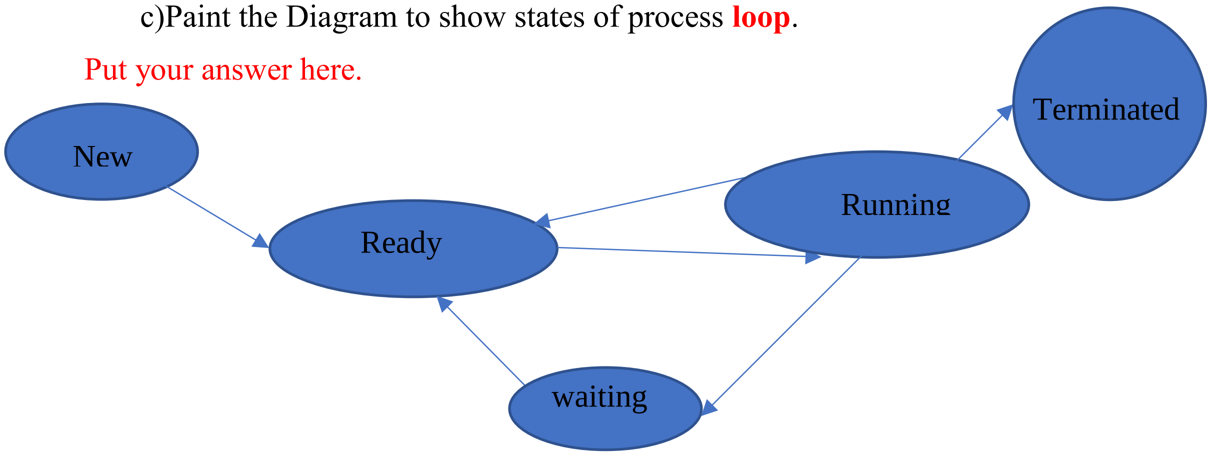
b)Which command do you use to terminate the **loop** program?

Put your answer here.

“./loop” Command + c
“./loop &” Command + d

c)Paint the Diagram to show states of process **loop**.

Put your answer here.



Problem 3.. Consider a process P, which needs to have 2 CPU-time units, 3 time units for I/O, and again 3 CPU-time units for its execution.

P:

CPU-time (2units)	I/O time (3units)	CPU-time (3units)

Let the time quantum allocated for a process in the running state be QT=1, and the process should be in the ready queue at least one time unit after entering in this state. The time of the process creation is QT=1. Continue the following time-state diagram:

State															
Running			*		*					*		*		*	
Ready		*		*					*		*		*		
Waiting						*	*	*							
New	*														
Terminated															*
Time →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Problem 4. Consider a process P from the problem 3. Let it produced the following child-process CH during one time unit after start:

CH:

CPU-time (1units)	I/O time (2units)	CPU-time (1units)

Make the time-state diagram for the processes P and CH for the following cases:

<u>Case 1</u>. Parent process P is waiting until the child-process CH will be finished

State	PROCESS P																				
Running			*									*					*		*		*	
Ready		*									*					*		*		*		
Waiting				*	*	*	*	*	*	*			*	*	*							
New	*																					
Terminated																						*
Time →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

State	PROCESS CH																				
Running						*				*												
Ready					*				*													
Waiting							*	*														
New				*																		
Terminated											*											
Time →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

<u>Case 2</u>. Parent process P is implementing concurrently with the child-process CH. Take, please, into account that the child process has higher priority.

State	PROCESS P																				
Running				*		*					*		*		*							
Ready		*	*		*					*		*		*								
Waiting							*	*	*													
New	*																					
Terminated																*						
Time →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

State	PROCESS CH																				
Running			*				*															
Ready		*				*																
Waiting				*	*																	
New	*																					
Terminated								*														
Time →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Problem 5. Answer the following questions

a) Explain the differences between a process and thread.

Put your answer here.

Process intended to be executed on the CPU. We can use any resources that control that the Linux kernel. Thread is executing units generated from a single process. We can start a thread in a parallel from the same process.

b) Consider a situation when the parent-process is terminated before than its dependent child-process was finished. What strategies can you suggest to resolve this situation? Please take into account that any child-process can also have its own child-processes.

Put your answer here.

Extermination→Child process forced termination.

Expiration→If processing didn't complete while setting time, it would finish child process.

Reincarnation →It sends time stamp to server(when client rebooting) and to finish child process..