

Assignment-2

Linux System and its Applications

Systems and Storage Laboratory

Department of Computer Science and Engineering

Chung-Ang University

Assignment-2

❖ Ctags :

1. Install ctags and read the manual (`$ man ctags`)
2. Take a screenshot of 'tags' file content

❖ Cscope :

1. Install cscope and read the manual (`$ man cscope`)
2. Take a screenshot of 'cscope.out' file content
 - You can replace this with the cscope execution screenshot

❖ Tmux :

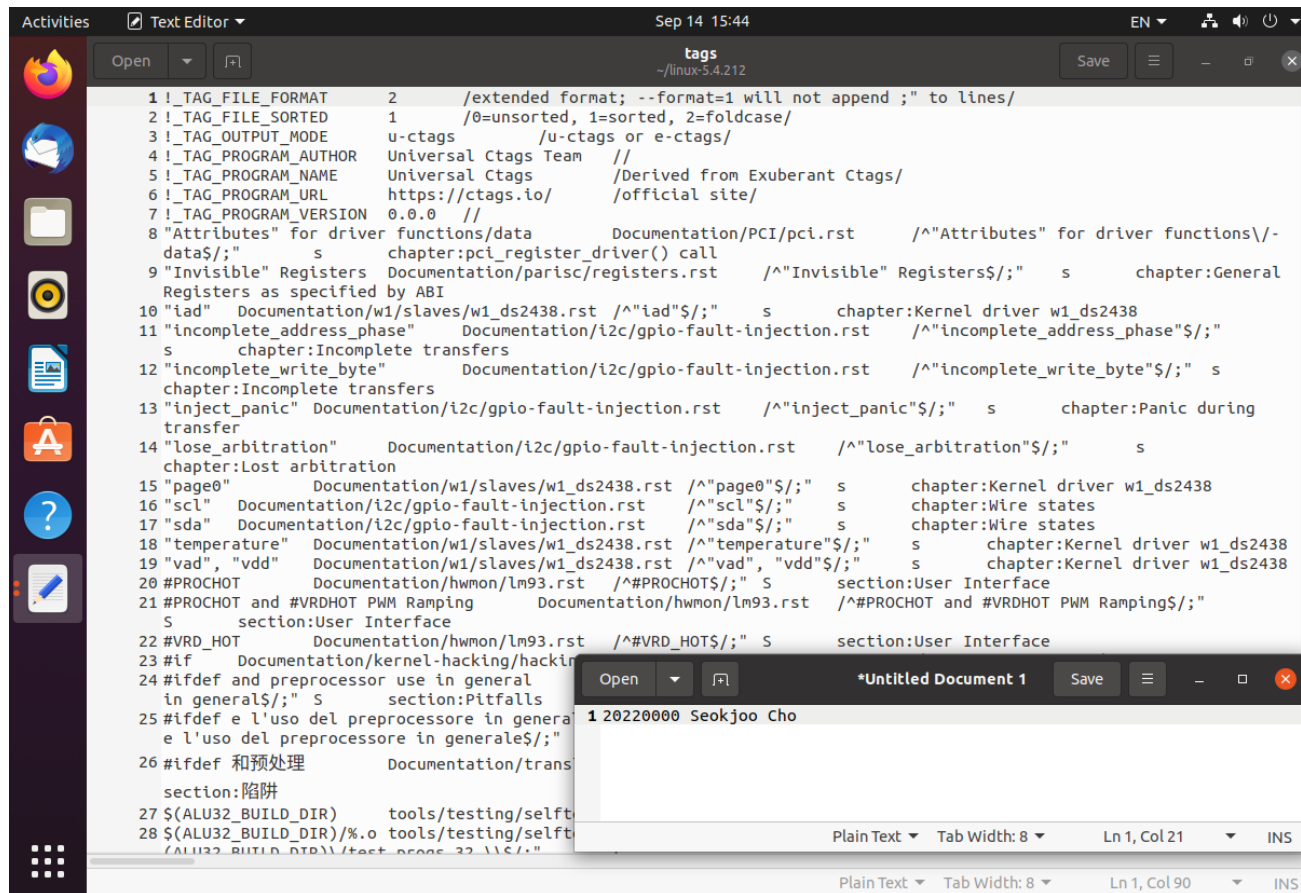
1. Install tmux and read the manual (`$ man tmux`)
2. **Window manipulation** screenshot
 - Create multiple panels

❖ Submit within **pdf** which includes all three screenshots

Ctags

❖ If you are using Ubuntu Desktop

- Generate tags file, open with your favorite editor (e.g. vim)
- Text Editor -> Your name with student id



The screenshot shows the Ubuntu Desktop environment. The main window is a Text Editor titled 'tags' with the path '~/.linux-5.4.212'. It contains a ctags file with the following content:

```
1 !_TAG_FILE_FORMAT      2       /extended format; --format=1 will not append ;" to lines/
2 !_TAG_FILE_SORTED      1       /0=unsorted, 1=sorted, 2=foldcase/
3 !_TAG_OUTPUT_MODE       u-ctags /u-ctags or e-ctags/
4 !_TAG_PROGRAM_AUTHOR    Universal Ctags Team //
5 !_TAG_PROGRAM_NAME       Universal Ctags       /Derived from Exuberant Ctags/
6 !_TAG_PROGRAM_URL       https://ctags.io/      /official site/
7 !_TAG_PROGRAM_VERSION   0.0.0 //
8 "Attributes" for driver functions/data      Documentation/PCI/pci.rst      /"Attributes" for driver functions\
data$/" s      chapter:pci_register_driver() call
9 "Invisible" Registers Documentation/parisc/registers.rst /"Invisible" Registers$/" s      chapter:General
Registers as specified by ABI
10 "iad" Documentation/w1/slaves/w1_ds2438.rst /"iad$/" s      chapter:Kernel driver w1_ds2438
11 "incomplete_address_phase" Documentation/i2c/gpio-fault-injection.rst /"incomplete_address_phase$/"
s      chapter:Incomplete transfers
12 "incomplete_write_byte" Documentation/i2c/gpio-fault-injection.rst /"incomplete_write_byte$/" s
chapter:Incomplete transfers
13 "inject_panic" Documentation/i2c/gpio-fault-injection.rst /"inject_panic$/" s      chapter:Panic during
transfer
14 "lose_arbitration" Documentation/i2c/gpio-fault-injection.rst /"lose_arbitration$/" s
chapter:Lost arbitration
15 "page0" Documentation/w1/slaves/w1_ds2438.rst /"page0$/" s      chapter:Kernel driver w1_ds2438
16 "scl" Documentation/i2c/gpio-fault-injection.rst /"scl$/" s      chapter:Wire states
17 "sda" Documentation/i2c/gpio-fault-injection.rst /"sda$/" s      chapter:Wire states
18 "temperature" Documentation/w1/slaves/w1_ds2438.rst /"temperature$/" s      chapter:Kernel driver w1_ds2438
19 "vdd" Documentation/w1/slaves/w1_ds2438.rst /"vdd$/" s      chapter:Kernel driver w1_ds2438
20 #PROCHOT Documentation/hwmon/lm93.rst /"PROCHOT$/" S      section:User Interface
21 #PROCHOT and #VRDHOT PWM Ramping Documentation/hwmon/lm93.rst /"PROCHOT and #VRDHOT PWM Ramping$/"
S      section:User Interface
22 #VRD_HOT Documentation/hwmon/lm93.rst /"VRD_HOT$/" S      section:User Interface
23 #if Documentation/kernel-hacking/hacking
24 #ifdef and preprocessor use in general
in general$/" S      section:Pitfalls
25 #ifdef e l'uso del preprocessore in generale
e l'uso del preprocessore in generale$/"
26 #ifdef 和预处理 Documentation/trans
section:陷阱
27 $(ALU32_BUILD_DIR) tools/testing/selft
28 $(ALU32_BUILD_DIR)/%o tools/testing/selft
(ALU32_BUILD_DIR)/test_page_22 \\\$/"
```

Overlaid on the bottom right is a smaller window titled '*Untitled Document 1' with the text '1 20220000 Seokjoo Cho'. The status bar at the bottom of the main window shows 'Plain Text', 'Tab Width: 8', 'Ln 1, Col 21', and 'INS'.

Ctags

❖ If you are using Ubuntu Server

- Generate tags file, open with your favorite editor (e.g. vim)
- Enter command mode -> Type your name and student id

```
1 !_TAG_FILE_FORMAT      2       /extended format; --format=1 will not append ;" to lines/
2 !_TAG_FILE_SORTED      1       /0=unsorted, 1=sorted, 2=foldcase/
3 !_TAG_OUTPUT_MODE      u-ctags /u-ctags or e-ctags/
4 !_TAG_PROGRAM_AUTHOR    Universal Ctags Team    //
5 !_TAG_PROGRAM_NAME      Universal Ctags /Derived from Exuberant Ctags/
6 !_TAG_PROGRAM_URL       https://ctags.io/        /official site/
7 !_TAG_PROGRAM_VERSION   0.0.0    //
8 "Attributes" for driver functions/data Documentation/PCI/pci.rst      /"Attributes" for d
river functions\data$%;" s      chapter:pci_register_driver() call
9 "Invisible" Registers Documentation/parisc/registers.rst      /"Invisible" Registers$%;"
s      chapter:General Registers as specified by ABI
10 "iad" Documentation/w1/slaves/w1_ds2438.rst /"iad"$%;" s      chapter:Kernel drive
r w1_ds2438
11 "incomplete_address_phase" Documentation/i2c/gpio-fault-injection.rst /"incomplet
e_address_phase"$%;" s      chapter:Incomplete transfers
12 "incomplete_write_byte" Documentation/i2c/gpio-fault-injection.rst /"incomplete_write_
byte"$%;" s      chapter:Incomplete transfers
13 "inject_panic" Documentation/i2c/gpio-fault-injection.rst /"inject_panic"$%;" s
chapter:Panic during transfer
14 "lose_arbitration" Documentation/i2c/gpio-fault-injection.rst /"lose_arbitration"
$%;" s      chapter:Lost arbitration
15 "page0" Documentation/w1/slaves/w1_ds2438.rst /"page0"$%;" s      chapter:Kernel drive
r w1_ds2438
16 "scl" Documentation/i2c/gpio-fault-injection.rst /"scl"$%;" s      chapter:Wire
states
17 "sda" Documentation/i2c/gpio-fault-injection.rst /"sda"$%;" s      chapter:Wire
states
18 "temperature" Documentation/w1/slaves/w1_ds2438.rst /"temperature"$%;" s      chap
ter:Kernel driver w1_ds2438
19 "vad", "vdd" Documentation/w1/slaves/w1_ds2438.rst /"vad", "vdd"$%;" s      chap
ter:Kernel driver w1_ds2438
20 #PROCHOT Documentation/hwmon/lm93.rst /"#PROCHOT"$%;" S      section:User Interfa
ce
21 #PROCHOT and #VRDHOT PWM Ramping Documentation/hwmon/lm93.rst /"#PROCHOT and #VRDH
OT PWM Ramping$%;" S      section:User Interface
```

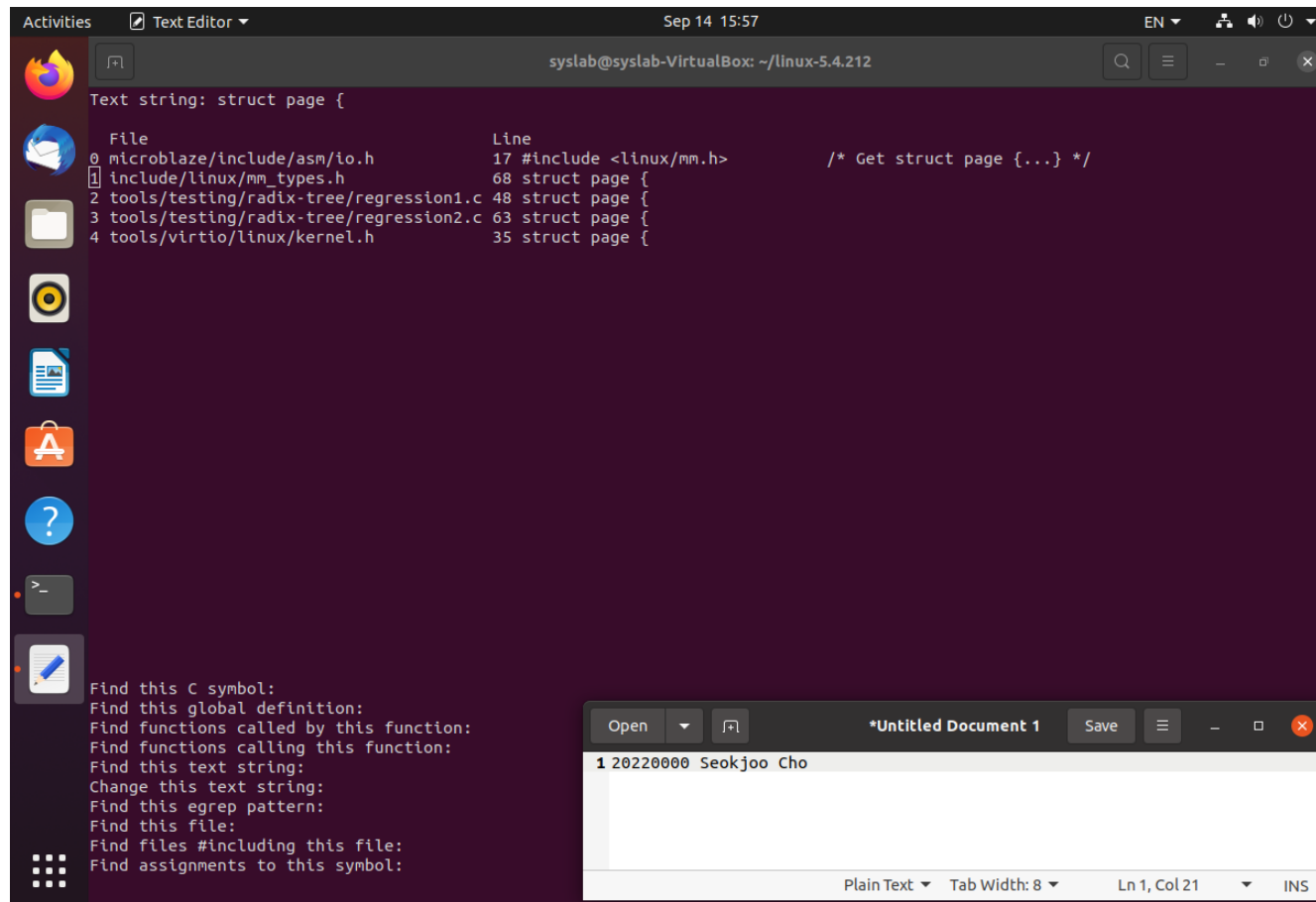
Your name
here, please!

```
###
:20220000 Seokjoo Cho
```

Cscope

❖ If you are using Ubuntu Desktop

- Generate cscope.out file, open with your favorite editor (e.g. vim)
 - Cscope execution screen is also good to go!
- Text Editor -> Your name with student id



Cscope

❖ For Ubuntu Server users, you have **two options**

1. Generate cscope.out file, open with your favorite editor
 - Enter command mode -> Type your name and student id
2. Cscope execution screen
 - Search menu -> Type your name and student id

```
1 cscope 15 $HOME/linux-5.4.212 0641358515
2 @Documentation/scheduler/sched-pelt.c
3
4 10 ^B
5 ~<m@h.h
6 >
7
8 11 ^B
9 ~<idio.h
10 >
11
12 13 ^A
13 #HALFLIFE
14 32
15
16 )
17
18 14 ^A
19 #SHIFT
20 32
21
22 )
23
24 16 ^H
25 gy
26 ;
27
28 18 ^^
29 $ÿlc_ruÂabE_avg_yN_<94v
30 (^)
31
32 20 ^R
33 i
34 ;
35
36 21 ^?
37 :20220000 Seokjoo Cho
```

Your name here, please!

or

Your name here, please!

```
Text string: struct pglist_data {
File Line
0 include/linux/mmzone.h 698 typedef struct pglist_data {

Find this C symbol:
Find this global definition:
Find functions called by this function:
Find functions calling this function:
Find this text string:
Change this text string:
Find this egrep pattern:
Find this file:
Find files #including this file:
Find assignments to this symbol: 20220000 Seokjoo Cho
```

Cscope

❖ Find the kernel functions by using cscope.

- Find **vfs_read**, and **vfs_write function bodies** (where the function is declared).
- Remind that you can search and explore functions by using ctags and cscope.

Tmux

❖ If you are using Ubuntu Desktop

- Create four panes as below.
- Text Editor -> Your name with student id.

```
syslab@syslab-server:~$ cat author.txt
20230000 YOUR_NAME
syslab@syslab-server:~$
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
32048	syslab	20	0	11532	3928	3056	S	1.2	0.1	0:00.46	tmux: server
32591	syslab	20	0	13264	4212	3424	R	1.2	0.1	0:00.41	top
32610	syslab	20	0	25988	10312	6716	S	1.2	0.1	0:00.16	vi
32619	syslab	20	0	25988	10292	6696	S	1.2	0.1	0:00.07	vi
1	root	20	0	169772	13044	8260	S	0.0	0.2	0:17.29	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	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_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0.0	0.0	0:00.08	ksoftirqd/0
10	root	20	0	0	0	0	I	0.0	0.0	0:12.70	rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.57	migration/0
12	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.72	migration/1

Tmux

❖ If you are using Ubuntu Server

- Create four panes as below.
- Print out a text file that contains your name and student id in one pane.

Your name
here, please!

```
450 ssize_t vfs_read(struct file *file, char __user *buf, size_t count, loff_t *pos)
451 {
452     ssize_t ret;
453
454     if (!(file->f_mode & FMODE_READ))
455         return -EBADF;
456     if (!(file->f_mode & FMODE_CAN_READ))
457         return -EINVAL;
458     if (unlikely(!access_ok(buf, count)))
459         return -EFAULT;
460
461     ret = rw_verify_area(READ, file, pos, count);
462     if (!ret) {
463         if (count > MAX_RW_COUNT)
464             count = MAX_RW_COUNT;
465         ret = __vfs_read(file, buf, count, pos);
466         if (ret > 0) {
467             fsnotify_access(file);
468             add_rchar(current, ret);
469         }
470         inc_syscr(current);
471     }
472
473     return ret;
474 }
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
syslab@syslab-server:~$ cat author.txt
20230000 YOUR_NAME
```

```
top - 14:40:43 up 1 day, 23:57, 5 users, load average: 0.15, 0.19, 0.08
%Cpu(s):  0.6 us,  0.3 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 7621.1 total, 134.7 free, 888.0 used, 6598.4 buff/cache
MiB Swap: 2048.0 total, 2048.0 free,  8.0 used, 6415.9 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 32048 syslab   20   0 11532   3928   3056 S   1.2   0.1   0:00.46 tmux: server
 32591 syslab   20   0 13264   4212  3424 R   1.2   0.1   0:00.41 top
 32610 syslab   20   0 25988  10312  6716 S   1.2   0.1   0:00.16 vi
 32619 syslab   20   0 25988  10292  6696 S   1.2   0.1   0:00.07 vi
    1 root      20   0 169772 13044   8260 S   0.0   0.2   0:17.29 systemd
    2 root      20   0      0      0      0 S   0.0   0.0   0:00.01 kthreadd
    3 root      20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root      20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root      20   0      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    8 root      20   0      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0      0      0      0 S   0.0   0.0   0:00.00 ksoftirqd/0
   10 root      20   0      0      0      0 I   0.0   0.0   0:12.70 rcu_sched
   11 root      rt    0      0      0      0 S   0.0   0.0   0:00.57 migration/0
   12 root     -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   14 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   16 root     -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
   17 root      rt    0      0      0      0 S   0.0   0.0   0:00.72 migration/1
```

Reminder of the submission details

❖ Submission details (all screenshots in **one pdf file**).

- Screenshots about ctags and cscope.
(please refer to the slide2; **three screenshots**).
- Screenshot of Final view
(please refer to the slides 8 & 9; **one screenshot**).
 - Which should include vfs_read, and vfs_write function body.

Screenshot 1

'tags' file
content

Screenshot 2

'cscope.out'
file content

Screenshot 3

Tmux Window
manipulation
screenshot

Screenshot 4

vfs_read function body.	vfs_write function body.
Your student ID and name.	Linux top command.