

lab4-3.

실습 과제

1) 디바이스 드라이버에 ioctl() 추가

- Device Driver [led_dd.c]

struct file_operations fops에 ioctl를 추가하니 오류가 발생하여, unlocked_ioctl을 추가하여 오류를 해결하였습니다. 오류의 원인은 커널 2.6.34 버전부터 기존의 ioctl이 사라지고, 대신 unlocked_ioctl로 대체되었기 때문입니다. BKL(Big Kernel Lock)로 보호되어 동작하는 ioctl은 SMP(Simultaneous Multi-Processing) 환경에서 불필요한 대기 시간을 발생시키는 비효율적인 면이 있어 이를 개선하고자 BKL을 사용하지 않는 unlocked_ioctl로 대체되었습니다.

따라서, unlocked_ioctl에 led_ioctl()을 등록해주었고, 이 함수에서 led의 색깔을 결정해주었습니다. application program의 argv[1] 입력값에 따라 전역변수 led_color의 값을 변경해주었습니다.

이후, led_write()에서 전역변수 led_color의 값에 따라 Red 또는 Green LED를 출력하도록 하였습니다.

```
1 #include <linux/kernel.h>
2 #include <linux/module.h>
3 #include <linux/init.h>
4 #include <linux/fs.h>
5 #include <linux/io.h>
6 #include <linux/delay.h>
7 #include <asm/io.h>
8 #include <asm/uaccess.h>
9
10 #define DRIVER_AUTHOR "KHU"
11 #define DRIVER_DESC "Led Driver"
12 #define DEVICE_NAME "led_dd"
13 #define MAJOR_NUMBER 222
14
15 #define GPIO_BASE 0x3F200000
16 #define GPIO_SIZE 0xC0
17
18 #define INPUT 0
19 #define OUTPUT 1
20 #define LOW 0
21 #define HIGH 1
22
23 void *iomem *gpioAddr;
24 volatile unsigned int gpioToGPSEL[] = {
25     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
26     4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
27     8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
28 };
29 volatile unsigned int gpioToShift[] = {
30     0, 3, 6, 9, 12, 15, 18, 21, 24, 27,
31     0, 3, 6, 9, 12, 15, 18, 21, 24, 27,
32     0, 3, 6, 9, 12, 15, 18, 21, 24, 27,
33 };
34 volatile unsigned int gpioToGPSET = 0x1C;
35 volatile unsigned int gpioToGPCLR = 0x28;
36
37 const int Led[16] = {
38     4, 17, 18, 27, 22, 23, 24, 25,
39     6, 12, 13, 16, 19, 20, 26, 21,
40 };
41
42 int led_color;
43
44 static volatile void initGpioAddr(void) {
45     if(gpioAddr == NULL) {
46         gpioAddr = ioremap(GPIO_BASE, GPIO_SIZE);
47     }
48 }
49
50 static void pinMode(int pin, int mode) {
51     volatile unsigned int *gpio = (volatile unsigned int *)gpioAddr;
52
53     unsigned int fsel = gpioToGPSEL[pin]/sizeof(unsigned int);
54     unsigned int shift = gpioToShift[pin];
55
56     if(mode) {
57         gpio[fsel] |= (1 << shift);
58     }
59     else {
60         gpio[fsel] |= (0 << shift);
61     }
62 }
63
64 static void digitalWrite(int pin, int value) {
65     volatile unsigned int *gpio = (volatile unsigned int *)gpioAddr;
66
67     unsigned int set = gpioToGPSET/sizeof(unsigned int);
68     unsigned int clr = gpioToGPCLR/sizeof(unsigned int);
69
70     if(value) {
71         gpio[set] = (1 << pin);
72     }
73     else {
74         gpio[clr] = (1 << pin);
75     }
76 }
77
78 static int led_open(struct inode *inode, struct file *filp) {
79     int i;
80
81     initGpioAddr();
82
83     for(i = 0; i < 16; i++) {
84         pinMode(Led[i], OUTPUT);
85         digitalWrite(Led[i], LOW);
86     }
87
88     printk("[led_dd] led_open\n");
89
90     return 0;
91 }
92
93 static int led_release(struct inode *inode, struct file *filp) {
94     iounmap(gpioAddr);
95
96     printk("[led_dd] led_realse\n");
97
98     return 0;
99 }
100
101 static int led_write(struct file *filp, const char *buf,
102                     size_t len, loff_t *f_pos) {
103     int i;
104     char state;
```

```

105 int led_pos;
106
107 if (led_color == 0)
108     led_pos = 3;
109 else if (led_color == 1)
110     led_pos = 11;
111
112 for(i = 0; i < len; i++) {
113     copy_from_user(&state, &buf[i], 1);
114
115     if(state == '0') {
116         digitalWrite(Led[led_pos-i], LOW);
117     }
118     else {
119         digitalWrite(Led[led_pos-i], HIGH);
120     }
121 }
122
123 printk("[led_dd] led_write\n");
124
125 return len;
126 }
127
128 static int led_ioctl(struct file *filep, unsigned int cmd, \
129                     unsigned long arg)
130 {
131     printk("[led_dd] led_ioctl() : set led_color to ");
132
133     if (arg == 0)
134     {
135         printk("Red\n");
136         led_color = 0;
137     }
138     else if (arg == 1)
139     {
140         printk("Green\n");
141         led_color = 1;
142     }
143
144     return led_color;
145 }
146
147 static struct file_operations fops = {
148     .owner          = THIS_MODULE,
149     .open           = led_open,
150     .release        = led_release,
151     .write          = led_write,
152     .unlocked_ioctl = led_ioctl,
153 };
154
155 static int led_init(void) {
156     printk("[led_dd] led init()\n");
157 }
158
159 #endif
160
161 #ifdef CONFIG_LEDS_GPIO
162 #endif
163
164 #endif
165
166 #endif
167
168 #endif
169
170 #endif
171
172 #endif
173
174 #endif
175
176 #endif
177
178 #endif
179
180 #endif
181
182 #endif
183
184 #endif
185
186 #endif
187
188 #endif
189
190 #endif
191
192 #endif
193
194 #endif
195
196 #endif
197
198 #endif
199
200 #endif
201
202 #endif
203
204 #endif
205
206 #endif
207
208 #endif
209
210 #endif
211
212 #endif
213
214 #endif
215
216 #endif
217
218 #endif
219
220 #endif
221
222 #endif
223
224 #endif
225
226 #endif
227
228 #endif
229
230 #endif
231
232 #endif
233
234 #endif
235
236 #endif
237
238 #endif
239
240 #endif
241
242 #endif
243
244 #endif
245
246 #endif
247
248 #endif
249
250 #endif
251
252 #endif
253
254 #endif
255
256 #endif
257
258 #endif
259
260 #endif
261
262 #endif
263
264 #endif
265
266 #endif
267
268 #endif
269
270 #endif
271
272 #endif
273
274 #endif
275
276 #endif
277
278 #endif
279
280 #endif
281
282 #endif
283
284 #endif
285
286 #endif
287
288 #endif
289
290 #endif
291
292 #endif
293
294 #endif
295
296 #endif
297
298 #endif
299
300 #endif
301
302 #endif
303
304 #endif
305
306 #endif
307
308 #endif
309
310 #endif
311
312 #endif
313
314 #endif
315
316 #endif
317
318 #endif
319
320 #endif
321
322 #endif
323
324 #endif
325
326 #endif
327
328 #endif
329
330 #endif
331
332 #endif
333
334 #endif
335
336 #endif
337
338 #endif
339
340 #endif
341
342 #endif
343
344 #endif
345
346 #endif
347
348 #endif
349
350 #endif
351
352 #endif
353
354 #endif
355
356 #endif
357
358 #endif
359
360 #endif
361
362 #endif
363
364 #endif
365
366 #endif
367
368 #endif
369
370 #endif
371
372 #endif
373
374 #endif
375
376 #endif
377
378 #endif
379
380 #endif
381
382 #endif
383
384 #endif
385
386 #endif
387
388 #endif
389
390 #endif
391
392 #endif
393
394 #endif
395
396 #endif
397
398 #endif
399
400 #endif
401
402 #endif
403
404 #endif
405
406 #endif
407
408 #endif
409
410 #endif
411
412 #endif
413
414 #endif
415
416 #endif
417
418 #endif
419
420 #endif
421
422 #endif
423
424 #endif
425
426 #endif
427
428 #endif
429
430 #endif
431
432 #endif
433
434 #endif
435
436 #endif
437
438 #endif
439
440 #endif
441
442 #endif
443
444 #endif
445
446 #endif
447
448 #endif
449
450 #endif
451
452 #endif
453
454 #endif
455
456 #endif
457
458 #endif
459
460 #endif
461
462 #endif
463
464 #endif
465
466 #endif
467
468 #endif
469
470 #endif
471
472 #endif
473
474 #endif
475
476 #endif
477
478 #endif
479
480 #endif
481
482 #endif
483
484 #endif
485
486 #endif
487
488 #endif
489
490 #endif
491
492 #endif
493
494 #endif
495
496 #endif
497
498 #endif
499
500 #endif
501
502 #endif
503
504 #endif
505
506 #endif
507
508 #endif
509
510 #endif
511
512 #endif
513
514 #endif
515
516 #endif
517
518 #endif
519
520 #endif
521
522 #endif
523
524 #endif
525
526 #endif
527
528 #endif
529
530 #endif
531
532 #endif
533
534 #endif
535
536 #endif
537
538 #endif
539
540 #endif
541
542 #endif
543
544 #endif
545
546 #endif
547
548 #endif
549
550 #endif
551
552 #endif
553
554 #endif
555
556 #endif
557
558 #endif
559
560 #endif
561
562 #endif
563
564 #endif
565
566 #endif
567
568 #endif
569
570 #endif
571
572 #endif
573
574 #endif
575
576 #endif
577
578 #endif
579
580 #endif
581
582 #endif
583
584 #endif
585
586 #endif
587
588 #endif
589
590 #endif
591
592 #endif
593
594 #endif
595
596 #endif
597
598 #endif
599
600 #endif
601
602 #endif
603
604 #endif
605
606 #endif
607
608 #endif
609
610 #endif
611
612 #endif
613
614 #endif
615
616 #endif
617
618 #endif
619
620 #endif
621
622 #endif
623
624 #endif
625
626 #endif
627
628 #endif
629
630 #endif
631
632 #endif
633
634 #endif
635
636 #endif
637
638 #endif
639
640 #endif
641
642 #endif
643
644 #endif
645
646 #endif
647
648 #endif
649
650 #endif
651
652 #endif
653
654 #endif
655
656 #endif
657
658 #endif
659
660 #endif
661
662 #endif
663
664 #endif
665
666 #endif
667
668 #endif
669
670 #endif
671
672 #endif
673
674 #endif
675
676 #endif
677
678 #endif
679
680 #endif
681
682 #endif
683
684 #endif
685
686 #endif
687
688 #endif
689
690 #endif
691
692 #endif
693
694 #endif
695
696 #endif
697
698 #endif
699
700 #endif
701
702 #endif
703
704 #endif
705
706 #endif
707
708 #endif
709
710 #endif
711
712 #endif
713
714 #endif
715
716 #endif
717
718 #endif
719
720 #endif
721
722 #endif
723
724 #endif
725
726 #endif
727
728 #endif
729
730 #endif
731
732 #endif
733
734 #endif
735
736 #endif
737
738 #endif
739
740 #endif
741
742 #endif
743
744 #endif
745
746 #endif
747
748 #endif
749
750 #endif
751
752 #endif
753
754 #endif
755
756 #endif
757
758 #endif
759
760 #endif
761
762 #endif
763
764 #endif
765
766 #endif
767
768 #endif
769
770 #endif
771
772 #endif
773
774 #endif
775
776 #endif
777
778 #endif
779
780 #endif
781
782 #endif
783
784 #endif
785
786 #endif
787
788 #endif
789
790 #endif
791
792 #endif
793
794 #endif
795
796 #endif
797
798 #endif
799
800 #endif
801
802 #endif
803
804 #endif
805
806 #endif
807
808 #endif
809
810 #endif
811
812 #endif
813
814 #endif
815
816 #endif
817
818 #endif
819
820 #endif
821
822 #endif
823
824 #endif
825
826 #endif
827
828 #endif
829
830 #endif
831
832 #endif
833
834 #endif
835
836 #endif
837
838 #endif
839
840 #endif
841
842 #endif
843
844 #endif
845
846 #endif
847
848 #endif
849
850 #endif
851
852 #endif
853
854 #endif
855
856 #endif
857
858 #endif
859
860 #endif
861
862 #endif
863
864 #endif
865
866 #endif
867
868 #endif
869
870 #endif
871
872 #endif
873
874 #endif
875
876 #endif
877
878 #endif
879
880 #endif
881
882 #endif
883
884 #endif
885
886 #endif
887
888 #endif
889
890 #endif
891
892 #endif
893
894 #endif
895
896 #endif
897
898 #endif
899
900 #endif
901
902 #endif
903
904 #endif
905
906 #endif
907
908 #endif
909
910 #endif
911
912 #endif
913
914 #endif
915
916 #endif
917
918 #endif
919
920 #endif
921
922 #endif
923
924 #endif
925
926 #endif
927
928 #endif
929
930 #endif
931
932 #endif
933
934 #endif
935
936 #endif
937
938 #endif
939
940 #endif
941
942 #endif
943
944 #endif
945
946 #endif
947
948 #endif
949
950 #endif
951
952 #endif
953
954 #endif
955
956 #endif
957
958 #endif
959
960 #endif
961
962 #endif
963
964 #endif
965
966 #endif
967
968 #endif
969
970 #endif
971
972 #endif
973
974 #endif
975
976 #endif
977
978 #endif
979
980 #endif
981
982 #endif
983
984 #endif
985
986 #endif
987
988 #endif
989
990 #endif
991
992 #endif
993
994 #endif
995
996 #endif
997
998 #endif
999
1000 #endif

```

```

157     register_chrdev(MAJOR_NUMBER, DEVICE_NAME, &fops);
158
159     return 0;
160 }
161
162 static void led_exit(void) {
163     printk("[led_dd] led_exit()\n");
164     unregister_chrdev(MAJOR_NUMBER, DEVICE_NAME);
165 }
166
167 module_init(led_init);
168 module_exit(led_exit);
169
170 MODULE_LICENSE("Dual BSD/GPL");

```

led_dd.c [x]
-- INSERT --

led_dd.c [x]
-- INSERT --

2) 해당 디바이스 드라이버를 이용하여 Red/Green LED 제어 application 작성

- Device Driver Application [digit_app2.c]

```
haseo@haseo-VirtualBox: ~/working/driver
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <string.h>
5 #include <sys/types.h>
6 #include <fcntl.h>
7 #include <linux/kdev_t.h>
8
9 #define _LED_PATH_ "/dev/led_dd"
10
11 int main(int argc, char **argv) {
12     int fd = 0;
13
14     if(argc != 3) {
15         printf("Usage: %s [LED color] [LED binary]\n", argv[0]);
16         exit(1);
17     }
18
19     if( (fd = open(_LED_PATH_, O_RDWR | O_NONBLOCK) ) < 0 ) {
20         perror("open()");
21         exit(1);
22     }
23
24     // set led color (0:RED, 1:GREEN)
25     ioctl(fd, 1, atol(argv[1]));
26
27     write(fd, argv[2], strlen(argv[2]));
28
29     close(fd);
30
31     return 0;
32 }
~
~
~
digit_app2.c 1,1
```

- Makefile

```
haseo@haseo-VirtualBox: ~/working/driver
1 MODULE_NAME=led_dd
2 KDIR=/home/haseo/working/linux/
3 PWD=$(shell pwd)
4 TARGET=arm
5 TOOLCHAIN=arm-linux-gnueabi-hf-
6 APP_NAME=digit_app2
7
8 obj-m:=$(MODULE_NAME).o
9
10 all: dd app
11
12 dd:
13     $(MAKE) -C $(KDIR) M=$(PWD) ARCH=$(TARGET) CROSS_COMPILE=$(TOOLCHAIN) modules
14
15 app:
16     arm-linux-gnueabi-hf-gcc -o $(APP_NAME) $(APP_NAME).c
17
18 clean:
19     rm -f *.ko
20     rm -f *.o
21     rm -f *.mod.*
22     rm -f modules.order
23     rm -f Module.symvers
24     rm -f *.cmd
25     rm -rf .tmp_versions
26     rm -f $(APP_NAME)
~
~
~
Makefile 1,1 All
```

- Insert Device Driver Module && Make Device File

```

pi@raspberrypi: ~/working
File Edit Tabs Help
pi@raspberrypi:~ $ cd working/
pi@raspberrypi:~/working $ ls -l
total 32
-rwxr-xr-x 1 pi pi 8476 Jun  4 13:16 digit_app
-rwxr-xr-x 1 pi pi 8548 Jun  4 15:54 digit_app2
-rw-r--r-- 1 pi pi 6704 Jun  4 15:54 led_dd.ko
pi@raspberrypi:~/working $ sudo insmod led_dd.ko
pi@raspberrypi:~/working $ lsmod | grep led_dd
led_dd                2408  0
pi@raspberrypi:~/working $ sudo mknod /dev/led_dd c 222 0
pi@raspberrypi:~/working $ sudo chmod 666 /dev/led_dd
pi@raspberrypi:~/working $ ls -l /dev/led_dd
crw-rw-rw- 1 root root 222, 0 Jun  4 15:57 /dev/led_dd
pi@raspberrypi:~/working $

```

- Execute Device Driver Application && Check Kernel Message

모든 LED를 OFF로 초기화했음에도 불구하고, LED15는 계속 ON 상태인 문제가 있습니다.
이는 이전 과제에서 발생한 문제와 동일한 하드웨어 문제인 것 같습니다.

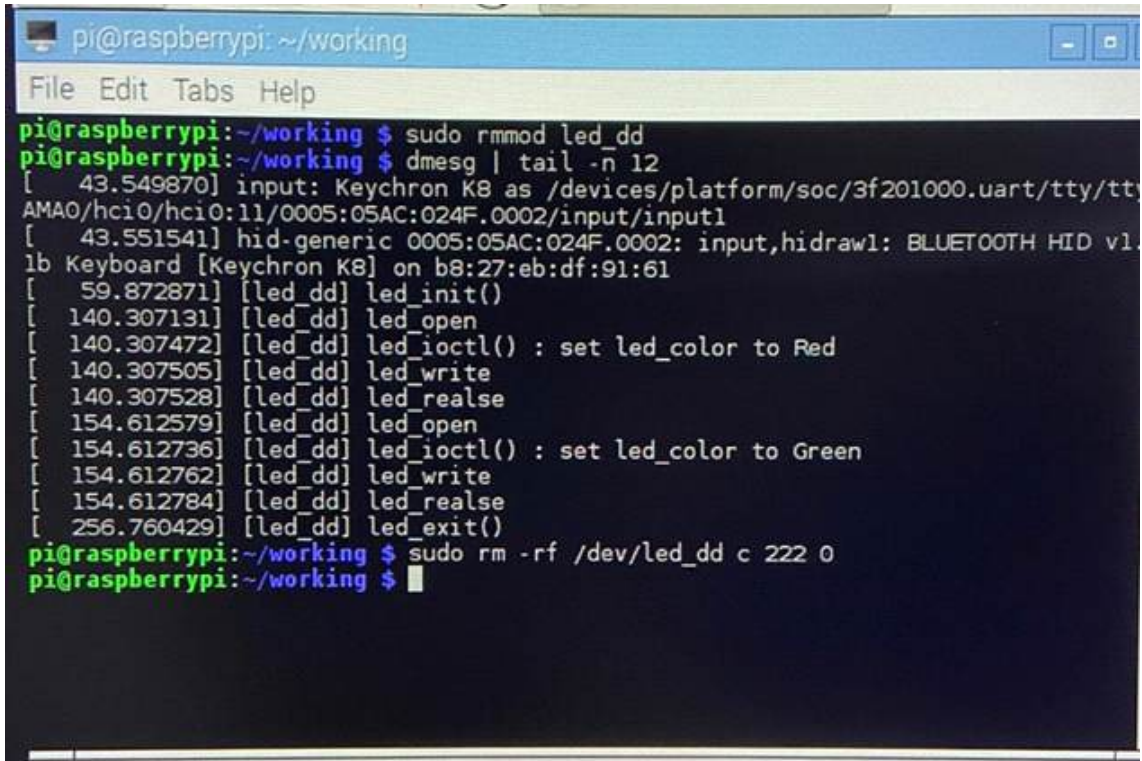
```

pi@raspberrypi: ~/working
File Edit Tabs Help
pi@raspberrypi:~/working $ ./digit_app2 0 1010
pi@raspberrypi:~/working $ ./digit_app2 1 1010
pi@raspberrypi:~/working $ dmesg | tail -n 10
[ 43.551541] hid-generic 0005:05AC:024F.0002: input,hidraw1: BLUETOOTH HID v1.
1
[ 59.872871] [led_dd] led_init()
[ 140.307131] [led_dd] led_open
[ 140.307472] [led_dd] led_ioctl() : set led_color to Red
[ 140.307505] [led_dd] led_write
[ 140.307528] [led_dd] led_realse
[ 154.612579] [led_dd] led_open
[ 154.612736] [led_dd] led_ioctl() : set led_color to Green
[ 154.612762] [led_dd] led_write
[ 154.612784] [led_dd] led_realse
pi@raspberrypi:~/working $

```



- Remove Device Driver Module && Remove Device File



```
pi@raspberrypi: ~/working
File Edit Tabs Help
pi@raspberrypi:~/working $ sudo rmmod led_dd
pi@raspberrypi:~/working $ dmesg | tail -n 12
[  43.549870] input: Keychron K8 as /devices/platform/soc/3f201000.uart/tty/tt
AMA0/hci0/hci0:11/0005:05AC:024F.0002/input/input1
[  43.551541] hid-generic 0005:05AC:024F.0002: input,hidraw1: BLUETOOTH HID v1
1b Keyboard [Keychron K8] on b8:27:eb:df:91:61
[  59.872871] [led_dd] led_init()
[ 140.307131] [led_dd] led_open
[ 140.307472] [led_dd] led_ioctl() : set led_color to Red
[ 140.307505] [led_dd] led_write
[ 140.307528] [led_dd] led_realise
[ 154.612579] [led_dd] led_open
[ 154.612736] [led_dd] led_ioctl() : set led_color to Green
[ 154.612762] [led_dd] led_write
[ 154.612784] [led_dd] led_realise
[ 256.760429] [led_dd] led_exit()
pi@raspberrypi:~/working $ sudo rm -rf /dev/led_dd c 222 0
pi@raspberrypi:~/working $
```