

임베디드시스템 실습 lab3

2017313107 이승태

emulator에 led와 7segment를 추가하고 qemu emulator로 돌려봤습니다.
syscall을 추가해 user-level에서 받은 숫자를 7segment에 출력했습니다.

- 새로운 사실: Ubuntu는 띄어쓰기를 할 때 _를 사용하는 것이 좋다.

아래의 사진은 323, 324번에 system call을 각각 추가해주고, 사용자에게 받은 인풋(char 형태)을 user_number, int로 바꾸어줍니다.

음수도 처리해주기 위해 -가 들어왔을 경우 user_number에 일괄적으로 -1을 넣어줍니다.
음수는 항상 0000000을 처리하도록 코딩해두었기 때문에 -1로 보내도 작동합니다.

그리고 __NR_segment를 호출할 때, user_number도 같이 parameter로 넣어주어 보내서 처리해줍니다.

```
1 #include <unistd.h>
2 #define __NR_segment 323
3 #define __NR_led 324
4
5 int main(int argc, char *argv[])
6 {
7     int user_number = 0;
8     if(argc >= 2)
9     {
10         int i = 0;
11         if(argv[1][0] == '-') user_number = -1;
12         else
13         {
14             while(argv[1][i] != '\0')
15             {
16                 user_number *= 10;
17                 user_number += argv[1][i] - 48;
18                 i++;
19             }
20         }
21     }
22
23     syscall(__NR_segment, user_number);
24     syscall(__NR_led);
25
26     return 0;
27 }
```

아래의 사진은 sys_segment가 segment_number을 인자로 받고 segment에 띄워줍니다.
number_array는 각 숫자가 7segment에 표시될 경우, 가지고 있는 숫자를 나타내줍니다.

만약 segment_number가 0보다 작거나 9999999보다 크다면 0000000을 띄웁니다. (37 ~ 46줄)

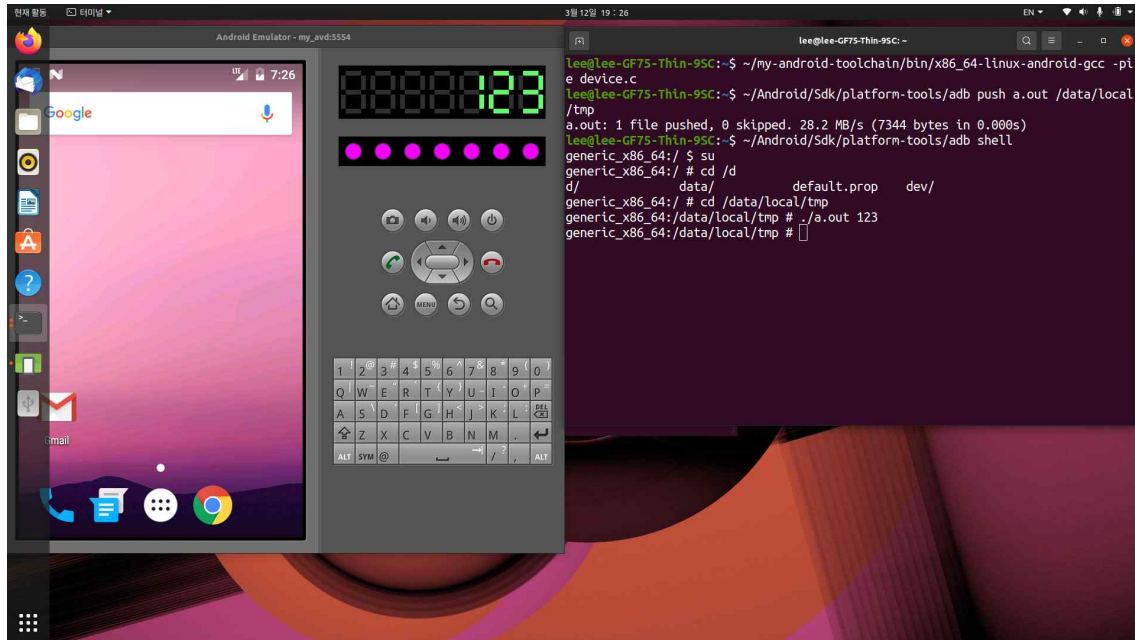
그렇지 않다면 오른쪽으로 정렬되게 숫자를 7segment에 출력합니다. (47 ~ 65줄)

위의 0000000과 구분하기 위해 0은 그냥 0만 띄우게 설정을 해주었습니다. (53, 54줄)

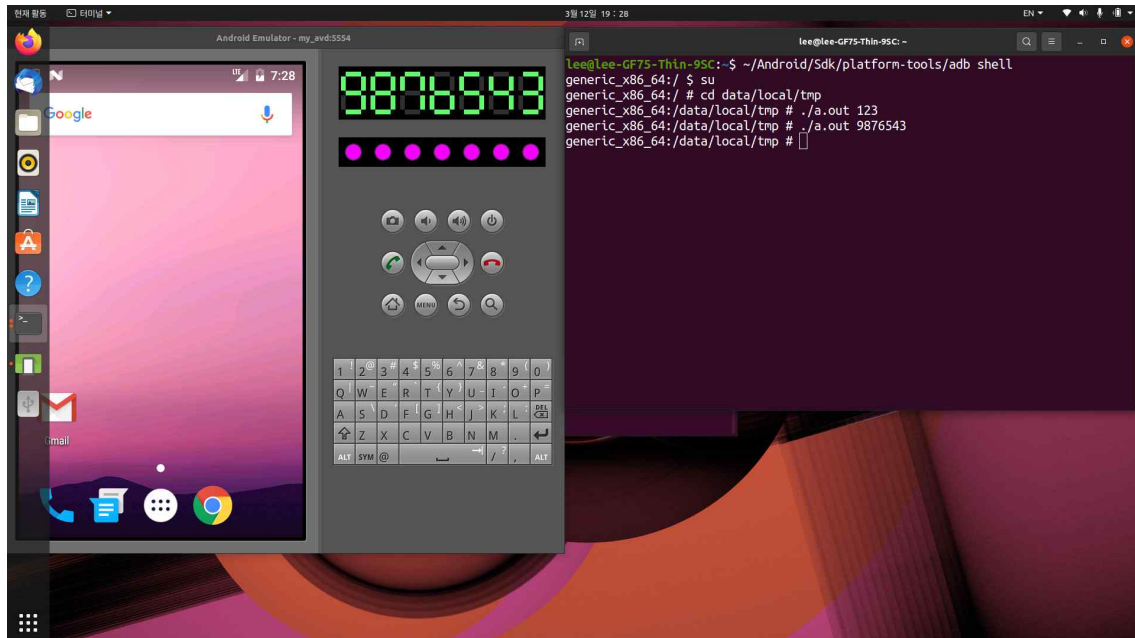
```
33 asmlinkage long sys_segment(int segment_number)
34 {
35     int number_array[10] = {0x77,0x24,0x5d,0x6d,0x2e,0x6b,0x7a,0x27,0x7f,0x6f};
36
37     if(segment_number > 9999999 || segment_number < 0)
38     {
39         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT0, number_array[0]);
40         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT1, number_array[0]);
41         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT2, number_array[0]);
42         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT3, number_array[0]);
43         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT4, number_array[0]);
44         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT5, number_array[0]);
45         GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT6, number_array[0]);
46     }
47     else
48     {
49         int i;
50         int SEGMENT_TMP = 0x018;
51         for(i=0;i<7;i++)
52         {
53             if(segment_number == 0 && i == 0)
54                 GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT_TMP, number_array[0]);
55
56             else if(segment_number == 0)
57                 GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT_TMP, 0x00);
58
59             else
60                 GOLDFISH_SEGMENT_WRITE(segment_data, SEGMENT_TMP, number_array[segment_number%10]);
61
62             segment_number /= 10;
63             SEGMENT_TMP -= 0x004;
64         }
65     }
66
67     return 0;
68 }
```

아래에 출력 예시가 있습니다.

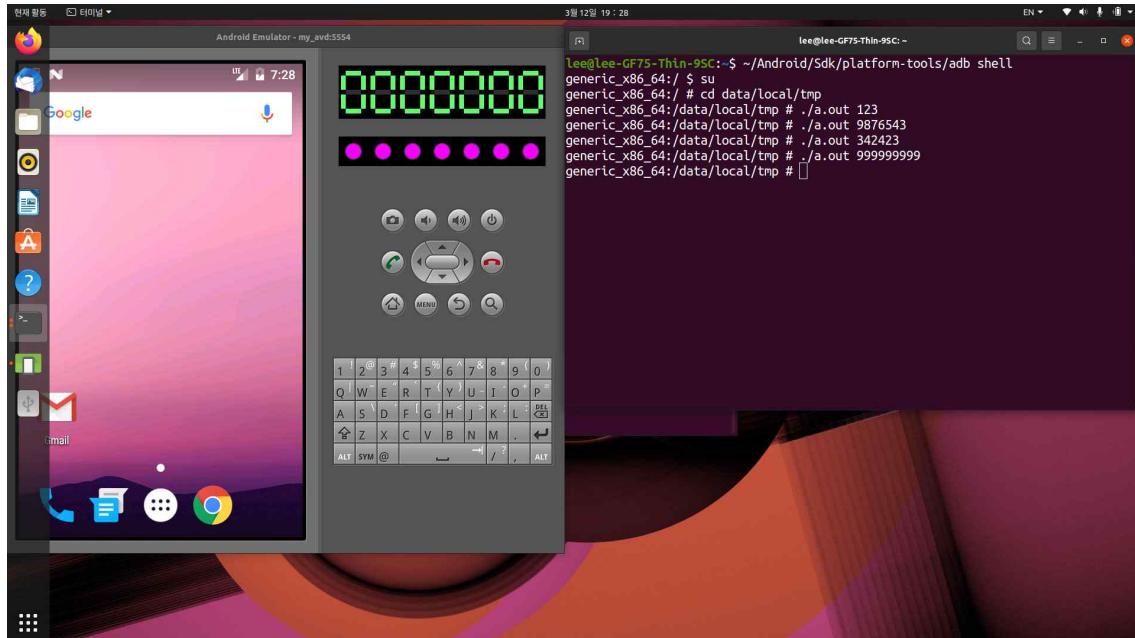
1. ./a.out 123을 넣은 모습



2. ./a.out 9876543을 넣은 모습



3. ./a.out 999999999 (9999999보다 큰값을 넣음)를 넣었을 경우



4. 0보다 작은 ./a.out -202를 넣은 모습

