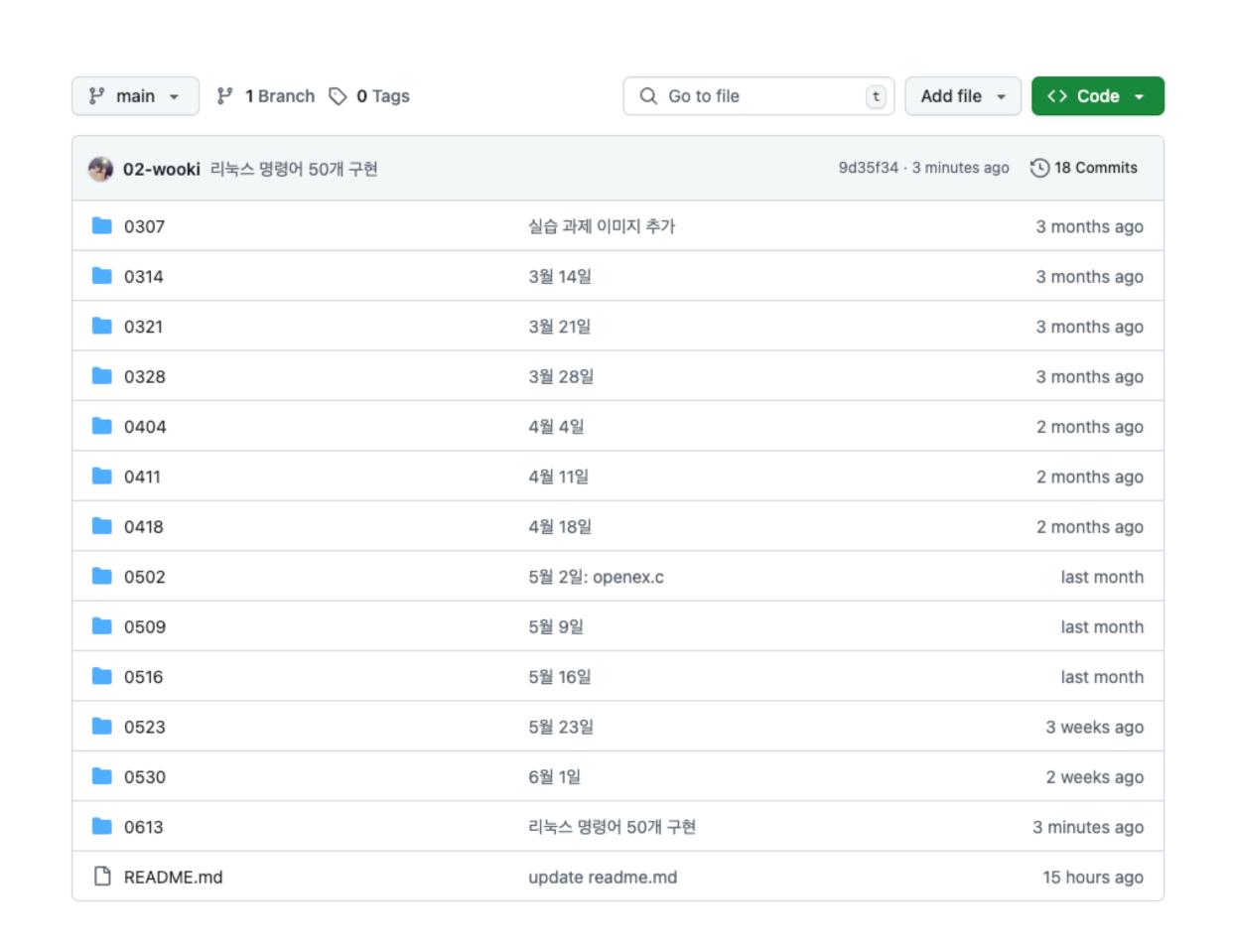
2025.06.13

깃허브 & 리눅스 명령어 구현

2025 시스템 프로그래밍

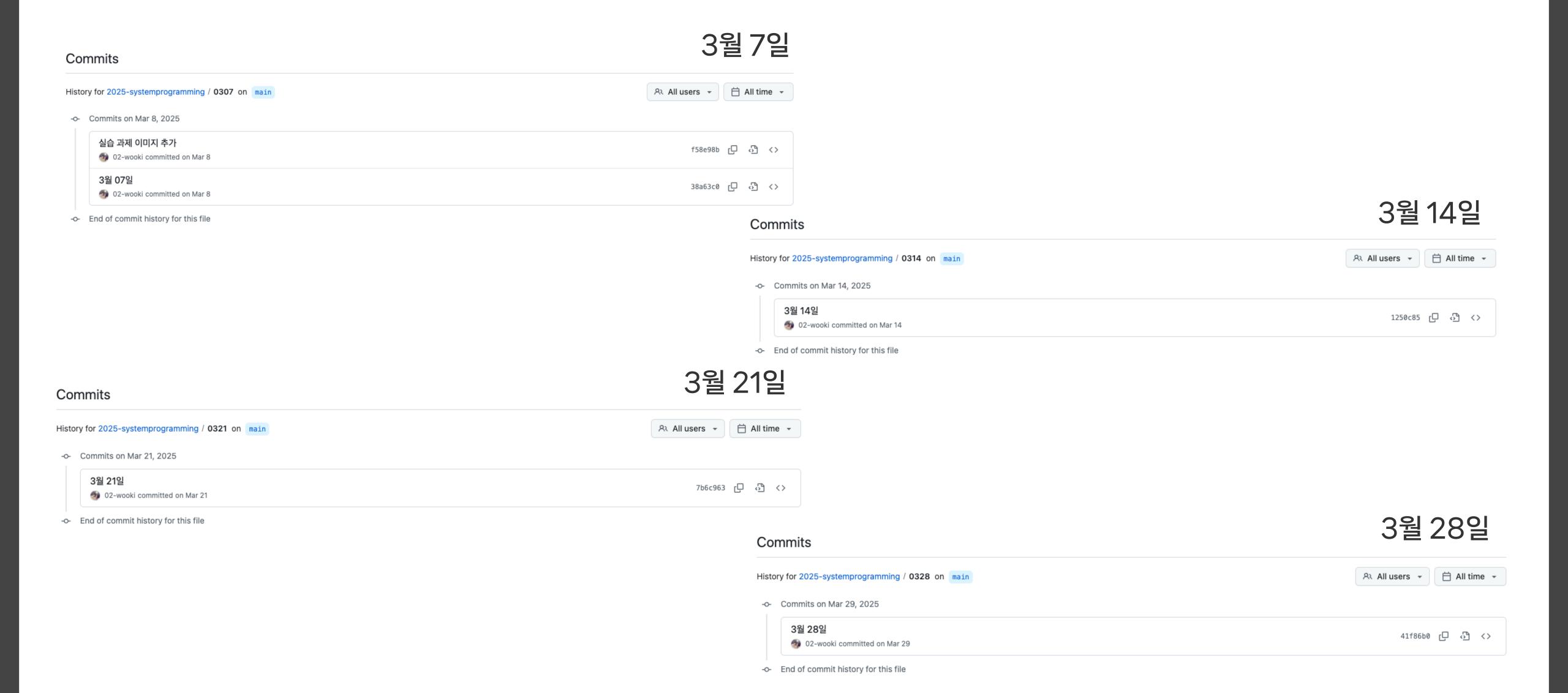
깃허브 리드미





깃허브 주소 https://github.com/02-wooki/2025-systemprogramming 리드미 제목 https://capsule-render.vercel.app/

깃허브 3월

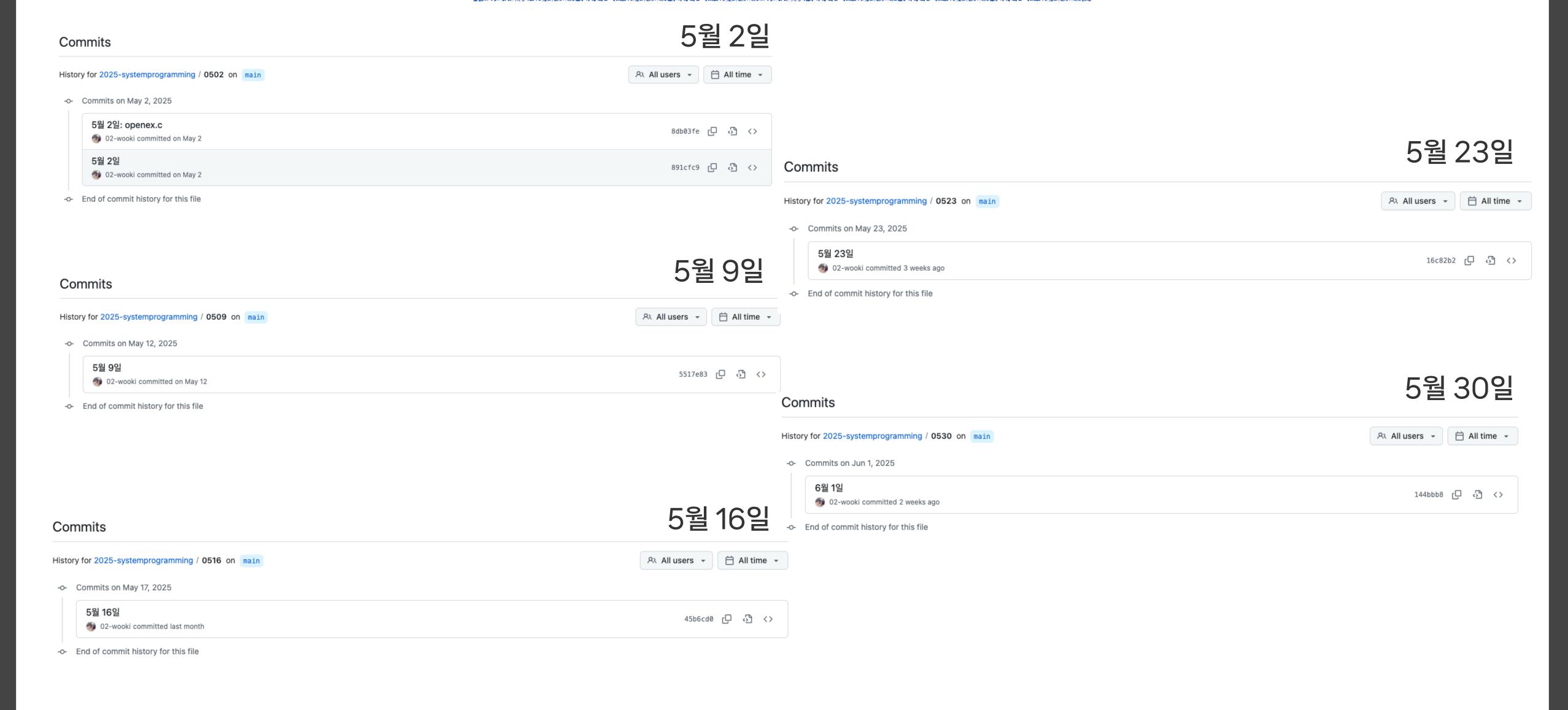


깃허브 4월

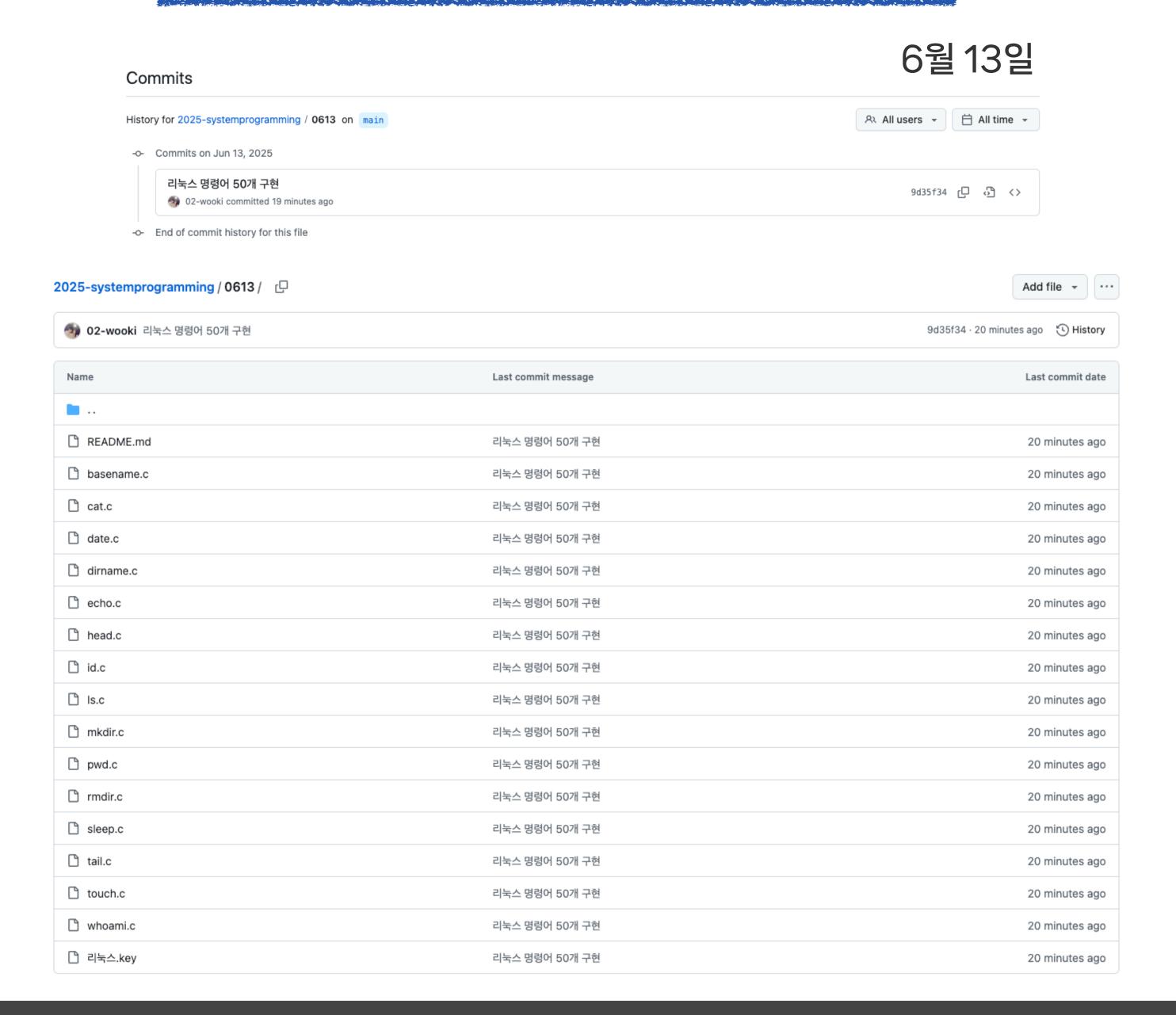
Commits	4월 4일
History for 2025-systemprogramming / 0404 on main	A All users → ⊟ All time →
-o- Commits on Apr 5, 2025	
4월 4일 ② 02-wooki committed on Apr 5	2b03c7a ☐ ③ 〈〉
-o- End of commit history for this file	
Commits	4월 11일
History for 2025-systemprogramming / 0411 on main	All users → ☐ All time →
-o- Commits on Apr 12, 2025	
4월 11일 ③ 02-wooki committed on Apr 12	ab6f18c ┌
-o- End of commit history for this file	
Commits	4월 18일
History for 2025-systemprogramming / 0418 on main	All users ▼ ⊟ All time ▼
-o- Commits on Apr 19, 2025	
4월 18일 ② 02-wooki committed on Apr 19	7018f87 ☐ ③ <>
> End of commit history for this file	

* 4월 25일은 중간고사

깃허브 5월



깃허브 6월



cat.c

리눅스의 cat.c 명령어를 C언어로 구현한 프로그램

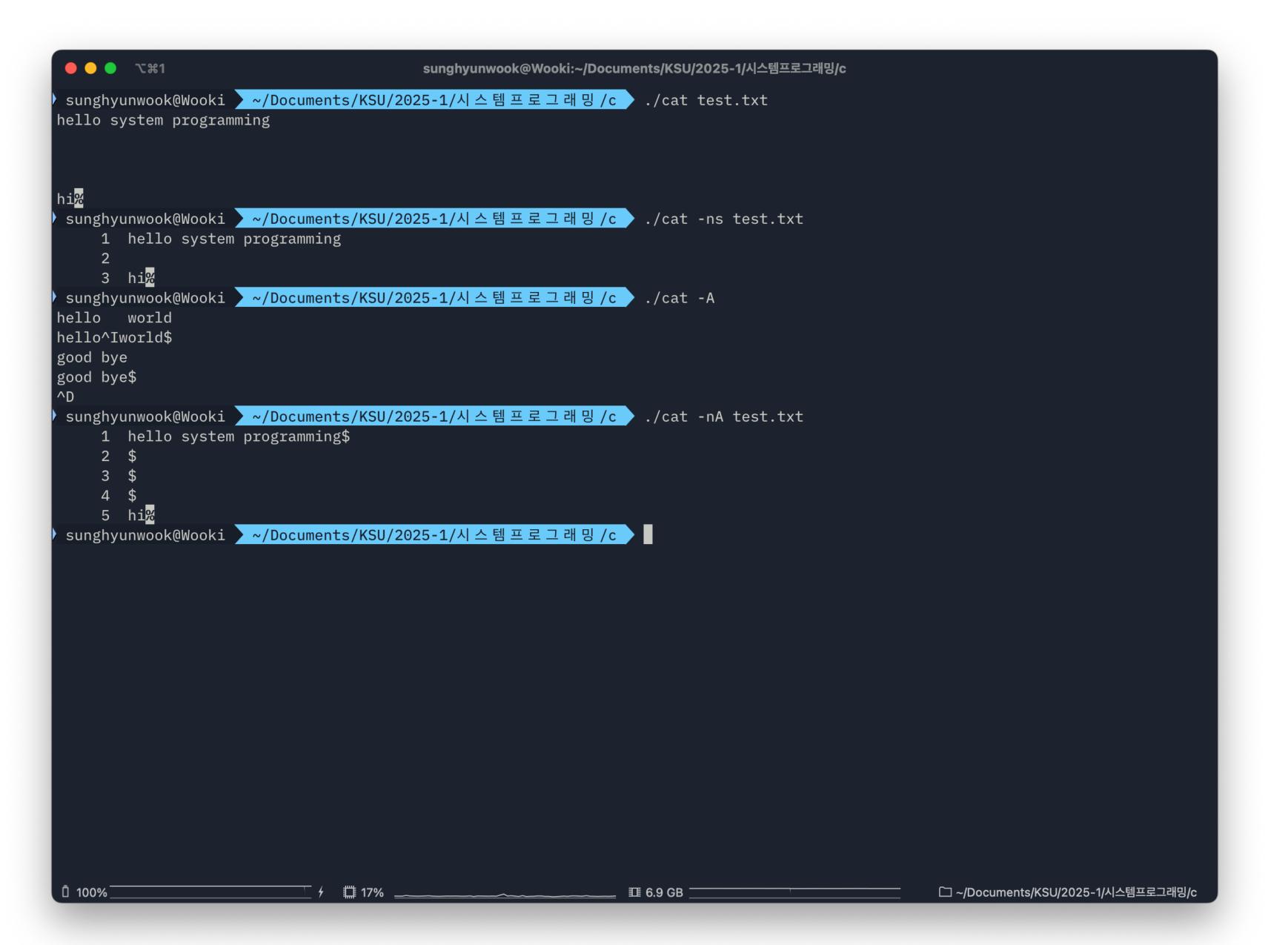
- -n 혹은 -number: 모든 줄에 번호 출력
- -b 혹은 -number-nonblank : 비어 있지 않은 줄에만 번호 출력
- -s 혹은 -squeez-blank: 연속된 빈 줄을 축약
- -E 혹은 -show-ends: 줄 끝에 \$ 기호 표시
- -A 혹은 -show-all: 제어 문자 및 줄 끝 기호 표시

```
1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #include <stdbool.h>
 4 #include <string.h>
 6 #define BUFFER_SIZE 1024
 8 // 옵션 플래그
 9 bool opt_b = false; // 번호 매기되 않은 빈 줄 제외하고 줄 번호 붙임
10 bool opt_n = false; // 모든 줄에 줄 번호 붙임
11 bool opt_s = false; // 연속된 빈 줄을 하나로 축소
12 bool opt_E = false; // 각 줄 끝에 $
13 bool opt_A = false; // -vET: 출력 불가능한 문자 시각화 + E + T 처리
15 int printFileWithOptions(FILE *fp) {
      char line[BUFFER_SIZE];
      int line_number = 1;
      bool prev_blank = false;
       while (fgets(line, BUFFER_SIZE, fp)) {
          bool is_blank = (line[0] = '\n');
           if (opt_s && is_blank && prev_blank)
              continue; // 연속된 빈 줄 제거
           prev_blank = is_blank;
           if (opt_b && is_blank = false) {
              printf("%6d\t", line_number++);
           } else if (opt_n && !opt_b) {
              printf("%6d\t", line_number++);
           for (int i = 0; line[i] \neq '\0' && line[i] \neq '\n'; ++i) {
              unsigned char c = line[i];
              if (opt_A) {
                  if (c = '\t')  {
                      printf("^I");
                  } else if (c < 32 && c \neq '\n') {
                      printf("^%c", c + 64);
                  } else if (c = 127) {
                      printf("^?");
                  } else {
                      putchar(c);
              } else {
                  putchar(c);
50
           if (opt_E || opt_A) {
              if (line[strlen(line) - 1] = '\n')
                  printf("$\n");
55
           } else {
              if (line[strlen(line) - 1] = '\n')
                  printf("\n");
59
     return 0;
62 }
```

```
63 int main(int argc, char *argv[]) {
       int file_start = 1;
        // 옵션 파싱
        for (int i = 1; i < argc; i++) {
            if (argv[i][0] \neq '-') {
                file_start = i;
                break;
            file_start = i + 1;
            if (argv[i][1] = '-') {
               int start = 2, end = strlen(argv[i]);
                char fullOpt[end - start + 1];
                for (int j = start; j < end; j++)
                   fullOpt[j - start] = argv[i][j];
                fullOpt[end - start] = '\0';
                if (!strcmp(fullOpt, "number-nonblank"))
                    opt_b = true;
                else if (!strcmp(fullOpt, "number"))
                    opt_n = true;
                else if (!strcmp(fullOpt, "squeeze-blank"))
                    opt_s = true;
                else if (!strcmp(fullOpt, "show-ends"))
                else if (!strcmp(fullOpt, "show-all"))
                    opt_A = true;
                    fprintf(stderr, "cat: invalid option -- '%s'\n", argv[i]);
               break;
            for (int j = 1; argv[i][j]; j++) {
                switch (argv[i][j]) {
                    case 'b': opt_b = true; break;
                    case 'n': opt_n = true; break;
                    case 's': opt_s = true; break;
                    case 'E': opt_E = true; break;
                    case 'A': opt_A = true; opt_E = true; break;
                        fprintf(stderr, "cat: invalid option -- '%c'\n", argv[i][j]);
                        exit(1);
        if (file_start ≥ argc) {
            printFileWithOptions(stdin);
            exit(0);
        for (int i = file_start; i < argc; i++) {</pre>
            FILE *fp = fopen(argv[i], "r");
                fprintf(stderr, "cat: %s: No such file or directory\n", argv[i]);
            printFileWithOptions(fp);
            fclose(fp);
        return 0;
```

cat.c

실행 화면



30점

기허브 늦게 올린 주차 없음 실습과제 누락 없음 리눅스 명령어 50개 구현 완료 README.md 작성

References

김주현 교수님 ftp 서버 자료들 https://ftpserver.ksecole.kr/

Chat GPT https://chatgpt.com/

C, Linux] Is 명령어 구현 - 개준생의 공부 일지, 티스토리 https://eteo.tistory.com/180

감사합니다

2021763040 성현욱