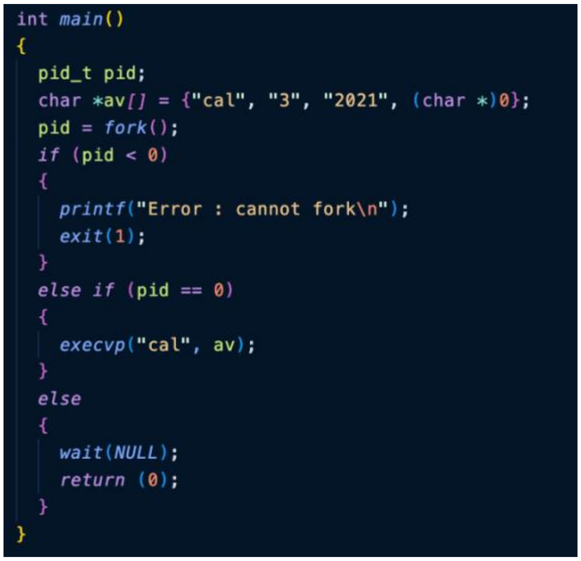
Activity 4

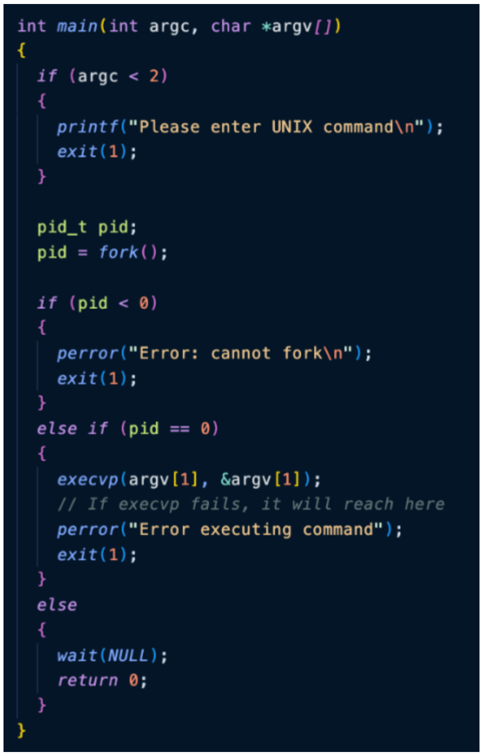
ธนัส วงศ์สมุทร 6432067021

ก้องภพ จริยาสถาพร 6430014321

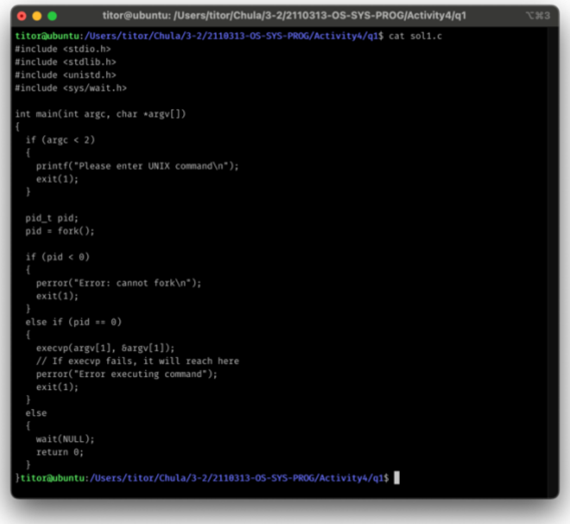
1. ทำการแก้ไข main ให้มีการรับ argument จากผู้ใช้ และนำไปใช้ใน execvp ต่อ



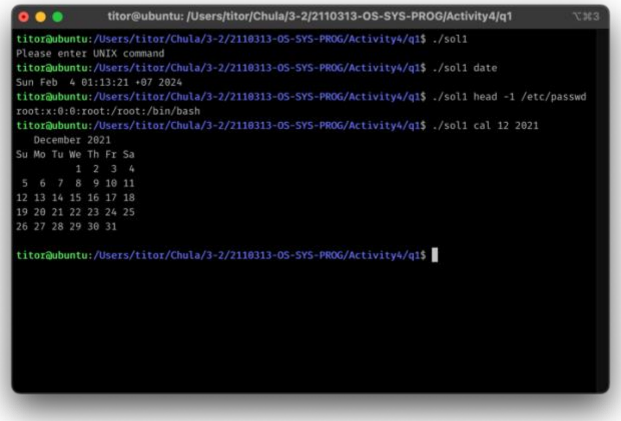
```c
int main()
{
  pid_t pid;
  char *av[] = {"cal", "3", "2021", (char *)0};
  pid = fork();
  if (pid < 0)
  {
    printf("Error : cannot fork\n");
    exit(1);
  }
  else if (pid == 0)
  {
    execvp("cal", av);
  }
  else
  {
    wait(NULL);
    return (0);
  }
}
```



```c
int main(int argc, char *argv[])
{
  if (argc < 2)
  {
    printf("Please enter UNIX command\n");
    exit(1);
  }

  pid_t pid;
  pid = fork();

  if (pid < 0)
  {
    perror("Error: cannot fork\n");
    exit(1);
  }
  else if (pid == 0)
  {
    execvp(argv[1], &argv[1]);
    // If execvp fails, it will reach here
    perror("Error executing command");
    exit(1);
  }
  else
  {
    wait(NULL);
    return 0;
  }
}
```

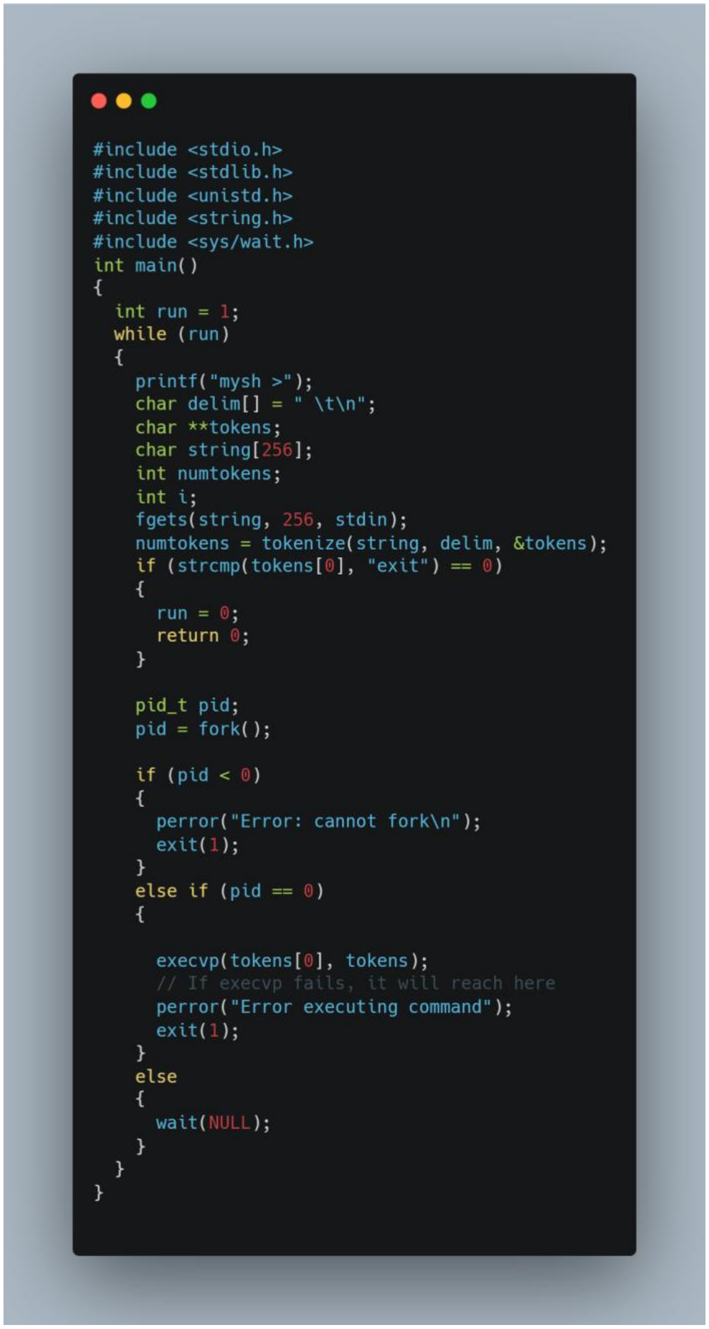

titor@ubuntu: /Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1
⌥⌘3

```
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$ cat sol1.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main(int argc, char *argv[])
{
  if (argc < 2)
  {
    printf("Please enter UNIX command\n");
    exit(1);
  }

  pid_t pid;
  pid = fork();

  if (pid < 0)
  {
    perror("Error: cannot fork\n");
    exit(1);
  }
  else if (pid == 0)
  {
    execvp(argv[1], &argv[1]);
    // If execvp fails, it will reach here
    perror("Error executing command");
    exit(1);
  }
  else
  {
    wait(NULL);
    return 0;
  }
}titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$
```

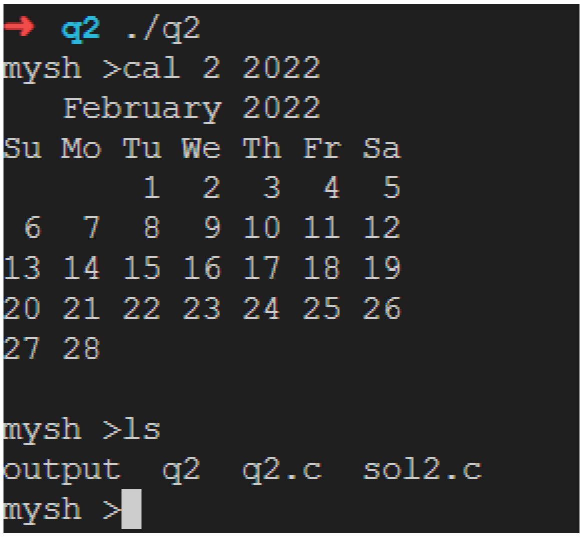

titor@ubuntu: /Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1
⌥⌘3

```
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$ ./sol1
Please enter UNIX command
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$ ./sol1 date
Sun Feb  4 01:13:21 +07 2024
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$ ./sol1 head -1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$ ./sol1 cal 12 2021
   December 2021
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity4/q1$
```

2. Similar to Q1, I check if proc is child and execvp and also check on parent whether if command is exit so it call exit directly on parent and break the while loop.



```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
int main()
{
  int run = 1;
  while (run)
  {
    printf("mysh >");
    char delim[] = " \t\n";
    char **tokens;
    char string[256];
    int numtokens;
    int i;
    fgets(string, 256, stdin);
    numtokens = tokenize(string, delim, &tokens);
    if (strcmp(tokens[0], "exit") == 0)
    {
      run = 0;
      return 0;
    }

    pid_t pid;
    pid = fork();

    if (pid < 0)
    {
      perror("Error: cannot fork\n");
      exit(1);
    }
    else if (pid == 0)
    {

      execvp(tokens[0], tokens);
      // If execvp fails, it will reach here
      perror("Error executing command");
      exit(1);
    }
    else
    {
      wait(NULL);
    }
  }
}
```



```
➜ q2 ./q2
mysh >cal 2 2022
   February 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28

mysh >ls
output  q2  q2.c  sol2.c
mysh >
```