**Activity 4: Simple Shell**

**สมาชิก**

|  |  |
| --- | --- |
| **ชื่อ-นามสกุล** | **เลขประจำตัวนิสิต** |
| นายเนติภัทร โพธิพันธ์ | 6631331621 |
| นายวรลภย์ ศรีชัยนนท์ | 6632200221 |
| นายสิปปภาส ชวานนท์ | 6630333721 |

1. **โปรแกรมข้างล่างแสดงการใช้ fork() และ execvp()  
   โปรเซสแม่สร้างโปรเซสลูก จากนั้นแม่คอยให้ลูกเรียก execvp() ทำงาน cal 3 2021**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<unistd.h>**

**#include<sys/wait.h>**

**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);**

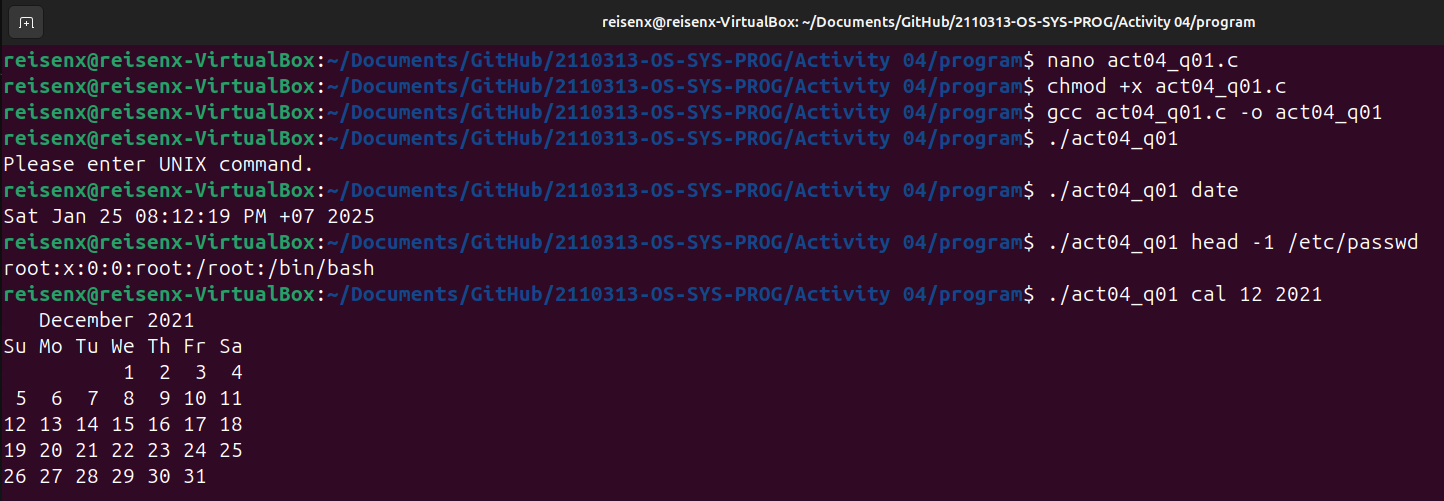
**return 0;**

**}**

**}**

**ขอให้นิสิตดัดแปลงโปรแกรมข้างบนให้รับ argument เป็นคำสั่ง LINUX แล้วทำงานคำสั่ง LINUX ตามที่ป้อน จากนั้นหยุดทำงาน  
ถ้าผู้ใช้ไม่ป้อน argument โปรแกรมจะบอกให้ผู้ใช้ป้อนคำสั่งและหยุดทำงาน**

****

****

1. **พิจารณาโปรแกรมข้างล่าง**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<unistd.h>**

**#include<string.h>**

**int main() {**

**/\* the program loops until exit \*/**

**int run = 1;**

**while(run) {**

**printf("mysh >");**

**/\***

**After reading user input, do these steps**

**1. use tokenize() function to get command**

**2. fork a child process**

**3. child use execvp() to run command**

**4. parent call wait() until user enter "exit"**

**\*/**

**}**

**}**

**/\***

**Split input string into substrings (called tokens)**

**which are sequences of contiguous characters separated by any of the character**

**in the string of accepted delimiters and fill the tokens into an array.**

**The last element of the array contains a NULL pointer.**

**Return number of tokens or -1 if not success**

**Example:**

**char delim[] = " \t\n";**

**char \*\*tokens;**

**char string[256];**

**int numtokens;**

**int i;**

**fget(string, 256, stdin);**

**numtokens = tokenize(string, delim, &tokens);**

**for(i = 0; i < numtokens; i++) {**

**printf("%d:%s\n", i, tokens[i]);**

**}**

**\*/**

**int tokenize(char \*string, char \*delimiters, char \*\*\*arrayOfTokens) {**

**char \*token;**

**int numtokens = 0;**

**int i;**

**/\* skip the beginning delimiters \*/**

**string += strspn(string, delimiters);**

**if((token = malloc(strlen(string) + 1)) == NULL) {**

**return -1;**

**}**

**/\* count tokens \*/**

**strcpy(token, string);**

**numtokens = 0;**

**if(strtok(token, delimiters) != NULL) {**

**for(numtokens = 1; strtok(NULL, delimiters) != NULL; numtokens++);**

**}**

**/\* create array of pointers to tokens \*/**

**if((\*arrayOfTokens = malloc((numtokens+1) \* sizeof(char \*))) == NULL) {**

**free(token);**

**return -1;**

**}**

**/\* fill pointers to tokens into the array \*/**

**if(numtokens == 0) free(token);**

**else {**

**strcpy(token, string);**

**(\*arrayOfTokens)[0] = strtok(token, delimiters);**

**for(i = 1; i < numtokens; i++) {**

**(\*arrayOfTokens)[i] = strtok(NULL, delimiters);**

**}**

**(\*arrayOfTokens)[numtokens] = NULL;**

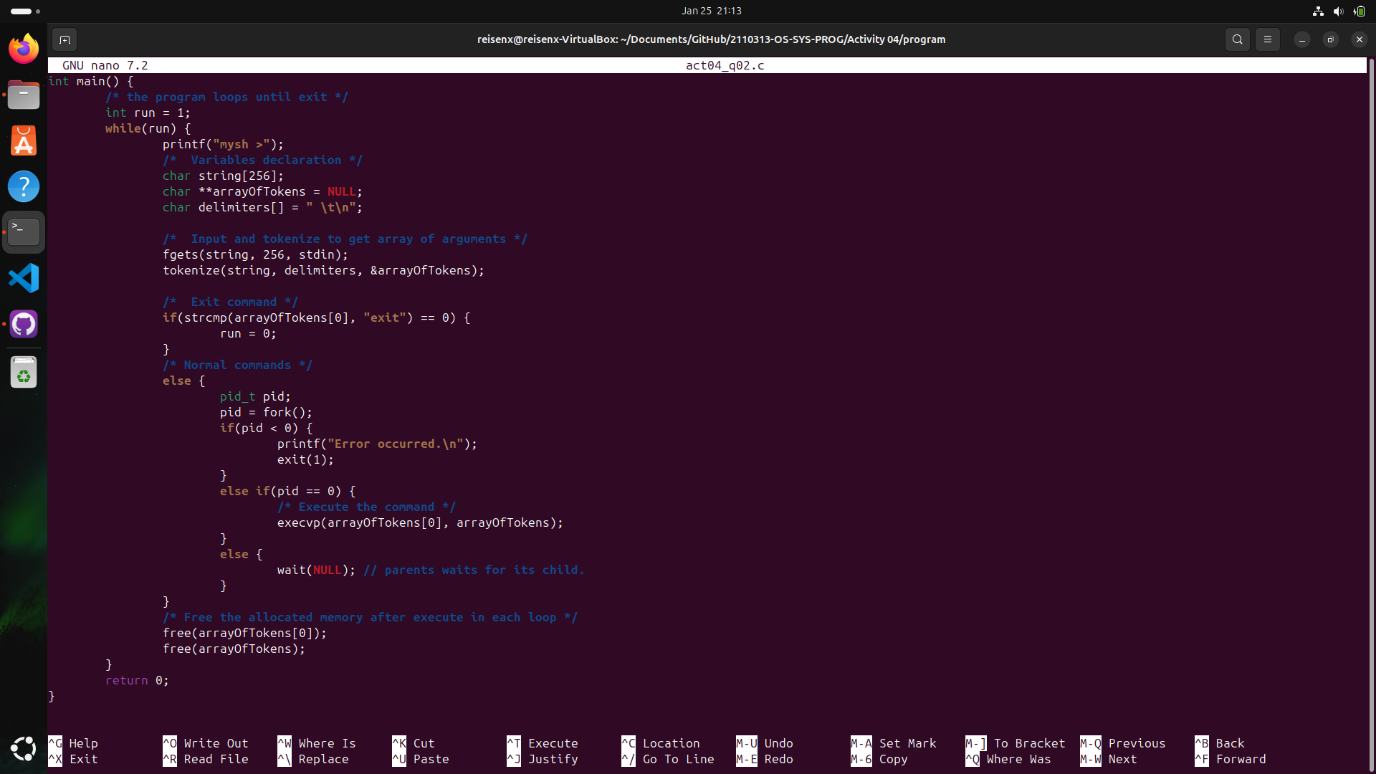
**}**

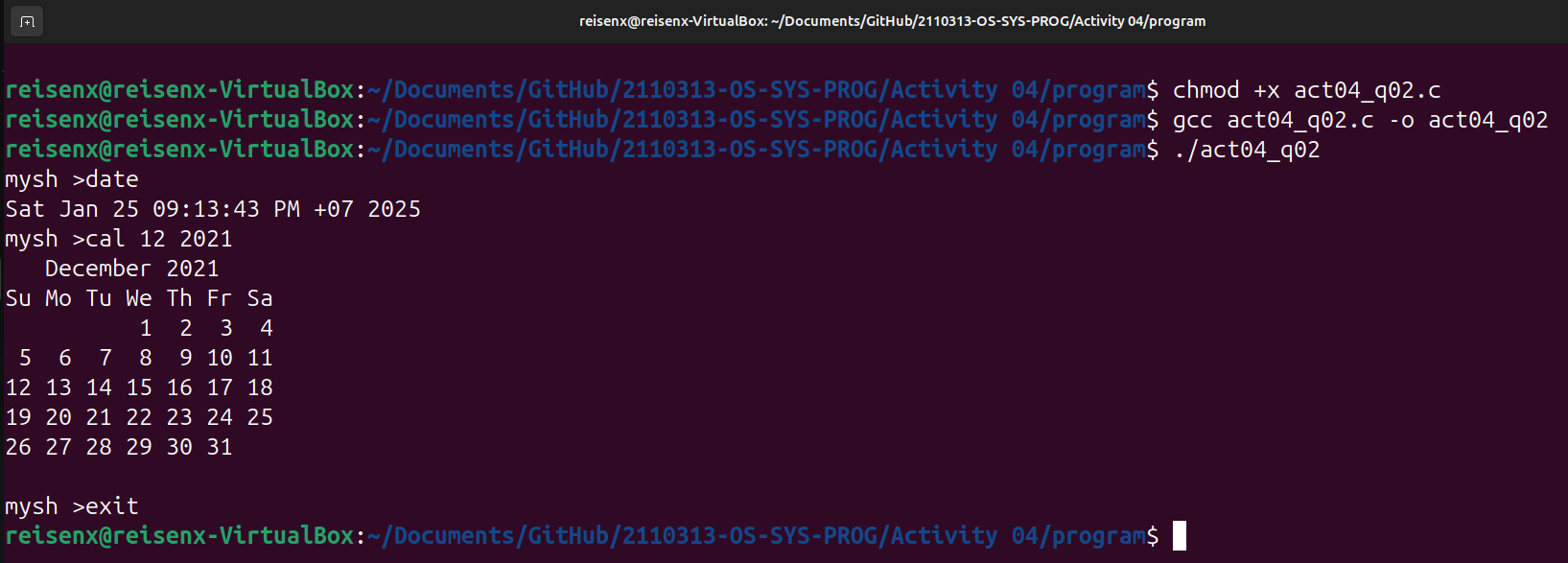
**return numtokens;**

**}**

**ใส่โค้ดใน main() เพื่อให้ทำงานเป็น shell แบบง่าย โดยที่ shell fork โปรเซสลูกเพื่อให้ลูกทำงานตามคำสั่ง LINUX ตามที่ผู้ใช้ป้อน และแม่ fork ลูกคอยรับคำสั่งใหม่เรื่อย ๆ จนกระทั่งผู้ใช้ป้อน exit จึงจะหยุดทำงาน  
Hint: ใช้ fgets() และฟังก์ชัน tokenize() เพื่อรับคำสั่งจากแป้นพิมพ์**

****

****

****