

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÁO CÁO HỆ ĐIỀU HÀNH (TN)
LAB 01:

Introduction to Linux Programming

Sinh viên thực hiện
Nguyễn Tấn Tài - 2212990

Thành phố Hồ Chí Minh – 2024

Exercise 3.6

- Which bash: /usr/bin/bash
- shebang: #! /usr/bin/bash

```
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212996$ which bash
/usr/bin/bash
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212996$ cat calc.sh
#!/usr/bin/bash

# Initialize variables
history=()

# Function to display history
display_history() {
    for entry in "${history[@]"; do
        echo "$entry"
    done
}

while true; do
    echo -n ">> "
    read input

    if [ "$input" = "EXIT" ]; then
        exit
    fi

    if [ "$input" = "HIS" ]; then
        display_history
        continue
    fi

    if [ "$input" = "ANS" ]; then
        echo <ANS.txt>
        continue
    fi

    operand1=$(echo "$input" | awk '{print $1}')
    operator=$(echo "$input" | awk '{print $2}')
    operand2=$(echo "$input" | awk '{print $3}')

    # Check for ANS for give to operand
    if [ "$operand1" = "ANS" ]; then
        operand1=$(cat ANS.txt)
    elif [ "$operand2" = "ANS" ]; then
        operand2=$(cat ANS.txt)
    fi
done
```

```
case $operator in
    +) res=$(echo "$operand1 + $operand2" | bc)
    ;;
    -) res=$(echo "$operand1 - $operand2" | bc)
    ;;
    *) res=$(echo "$operand1 * $operand2" | bc)
    ;;
    /)
        # Devision operation
        if [ "$operand2" -eq 0 ]; then
            res="MATH ERROR"
        else
            res=$(echo "scale =2;$operand1 / $operand2" | bc)
        fi
    ;;
    %) res=$(echo "$operand1 % $operand2" | bc)
    ;;
    *) res="SYNTAX ERROR"
    ;;
esac
if [ "$res" != "SYNTAX ERROR" ] && [ "$res" != "MATH ERROR" ] && [ "$input" != "HIS" ]; then
    echo "$res" > ANS.txt
    history+=("$input = $res")
    if [ ${#history[@]} -gt 5 ]; then
        history=( "${history[@]: -5}" )
    fi
fi
echo "$res"
read -n1 -s -r
clear
done
```

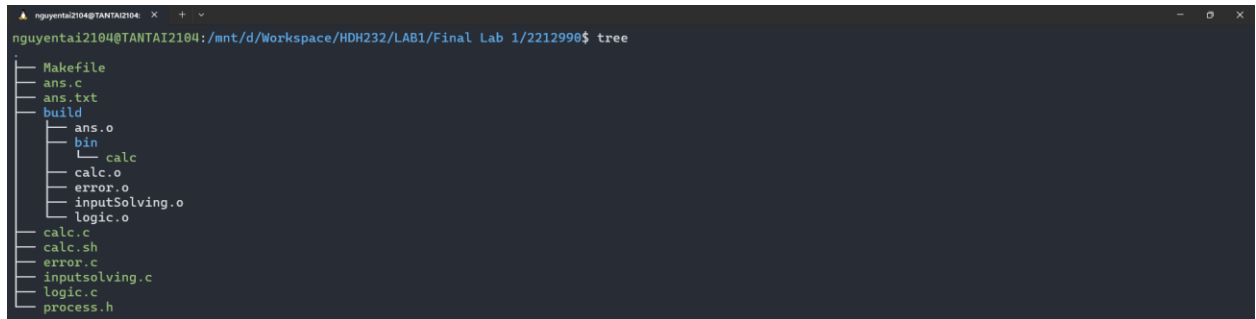
- Để thuận tiện hơn trong việc chụp ảnh screenshots, em xin phép được comment dòng clear – có chức năng sẽ xóa màn hình terminal sau khi “press any key to continue”



```
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ bash calc.sh
>> 1 + 2
3
>> 5 / 3
1.66
>> 1 / 0
MATH ERROR
>> 1 0 2
SYNTAX ERROR
>> 2 + 5
7
>> ANS + 3
10
>> 3 x 5
15
>> 9 / 6
1.50
>> 25 / 6
4.16
>> 9 % 6
3
>> HIS
ANS + 3 = 10
3 x 5 = 15
9 / 6 = 1.50
25 / 6 = 4.16
9 % 6 = 3
>> HS
SYNTAX ERROR
>> TEMP
SYNTAX ERROR
>> EXIT
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ |
```

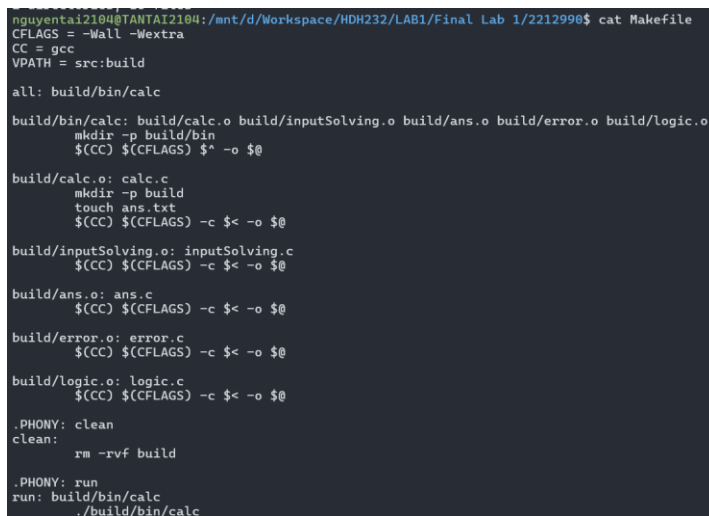
Exercise 5.3

- Đây là biểu diễn kiểu sơ đồ cây của bài tập 5.3



```
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ tree
.
├── Makefile
├── ans.c
├── ans.txt
├── build
│   ├── ans.o
│   ├── bin
│   │   └── calc
│   ├── calc.o
│   ├── error.o
│   ├── inputSolving.o
│   └── logic.o
├── calc.c
├── calc.sh
├── error.c
├── inputSolving.c
├── logic.c
└── process.h
```

- Makefile



```
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ cat Makefile
CFLAGS = -Wall -Wextra
CC = gcc
VPATH = src:build

all: build/bin/calc

build/bin/calc: build/calc.o build/inputSolving.o build/ans.o build/error.o build/logic.o
    mkdir -p build/bin
    $(CC) $(CFLAGS) $^ -o $@

build/calc.o: calc.c
    mkdir -p build
    touch ans.txt
    $(CC) $(CFLAGS) -c $< -o $@

build/inputSolving.o: inputSolving.c
    $(CC) $(CFLAGS) -c $< -o $@

build/ans.o: ans.c
    $(CC) $(CFLAGS) -c $< -o $@

build/error.o: error.c
    $(CC) $(CFLAGS) -c $< -o $@

build/logic.o: logic.c
    $(CC) $(CFLAGS) -c $< -o $@

.PHONY: clean
clean:
    rm -rvf build

.PHONY: run
run: build/bin/calc
    ./build/bin/calc
```

Trong đó:

- + VPATH: là đường dẫn chỉ định cho nơi chứa các file được tạo ra khi run các files
- + .PHONY: thuộc tính cho phép chạy ngay target có tên được đặt ra mà không cần bước phải kiểm tra sự tồn tại của file có sự trùng lặp tên với target đó (cụ thể ở đây là clean and run)
- + \$^: đề cập tới tất cả các dependencies của target đó

+ \$<: chỉ đề cập tới dependency đầu tiên của target đó

+ \$@: file được tạo ra sẽ lấy tên của target

- File calc.c

```
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212998$ cat calc.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "process.h"

#define MAX_LENGTH 200

int main()
{
    while (1)
    {
        printf(">>> ");

        // Input Processing
        char input[MAX_LENGTH];
        fgets(input, MAX_LENGTH, stdin);
        if (input[strlen(input) - 1] == '\n') // If have a newline character at the last => remove
            input[strlen(input) - 1] = '\0';

        // OUT POINT of the while loop
        if (strcmp(input, "EXIT") == 0)
            return 0;

        if (strcmp(input, "ANS") == 0)
        {
            printf("%.2lf\n", getAns());
            clearScreenAndPrompt();
            continue;
        }

        // Solve Input String
        double num1, num2;
        char op;
        if (!solveInputString(input, &num1, &num2, &op))
        {
            printf("SYNTAX ERROR\n");
            clearScreenAndPrompt();
            continue;
        }
    }
}
```

```
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212998$
    printf("%.2lf\n", getAns());
    clearScreenAndPrompt();
    continue;
}

// Solve Input String
double num1, num2;
char op;
if (!solveInputString(input, &num1, &num2, &op))
{
    printf("SYNTAX ERROR\n");
    clearScreenAndPrompt();
    continue;
}

// Check cases
if (isMathError(&op, &num2))
{
    printf("MATH ERROR\n");
    clearScreenAndPrompt();
    continue;
}
if (isSyntaxError(&op))
{
    printf("SYNTAX ERROR\n");
    clearScreenAndPrompt();
    continue;
}

// Here, Math is VALID ==> calculate
double ans = calculationLogic(&num1, &num2, &op);

// Print ans to screen
printf("%.2lf\n", ans);

// Update this success math answer to ans.txt
updateAns(ans);

// Wait for another input ==> Clear the screen
clearScreenAndPrompt();
}

return 0;
nguyentai2104@TANTA12104: /mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212998$
```

- File process.h: là file header chứa tất cả các functions sẽ được thực thi bởi nhiều file c khác, phụ vụ cho hàm calc.c

```
nguyentai2104@TANTA12104:/mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ cat process.h
#include <stdio.h>
#include <stdlib.h> // for atof
#include <string.h> // for strstr
#include <unistd.h> // for readlink
#include <libgen.h> // for dirname
#include <errno.h>
#include <stdbool.h>

#define ANSFILE "ans.txt"
#define MAX_NUMBER_SIZE 16

void clearScreenAndPrompt();
bool isNumeric(const char *str);
bool solveInputString(char *input, double *num1, double *num2, char *op);

bool isSyntaxError(char *op);
bool isMathError(char *op, double *num2);

double calculationLogic(double *num1, double *num2, char *op);
double addition(double *num1, double *num2);
double subtraction(double *num1, double *num2);
double multiplication(double *num1, double *num2);
double division(double *num1, double *num2);
double modulo(double *num1, double *num2);

void updateAns(double newAns);
double getAns();
```

- File error.c, inputsolving.c, logic.c là các hàm C thực thi các functions được liệt kê ở header theo từng mục đích được đặt ra

```
nguyentai2104@TANTA12104:/mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ cat error.c
#include "process.h"

bool isSyntaxError(char *op){
    if(*op == '+' || *op == '-' || *op == 'x' || *op == '/' || *op == '%')
        return false;
    return true;
}

bool isMathError(char *op, double *num2){
    if((*num2 == 0 && *op == '/') || (*num2 == 0 && *op == '%'))
        return true;
    return false;
}
nguyentai2104@TANTA12104:/mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ |
```

```
nguyentai2104@TANTA12104:/mnt/d/Workspace/HDH232/LAB1/Final Lab 1/2212990$ cat inputSolving.c
#include "process.h"

void clearScreenAndPrompt()
{
    fflush(stdout);
    getchar(); // Get only one character
    system("clear || cls");
}

bool isNumeric(const char *str) {
    char *endptr;
    strtod(str, &endptr);
    return *endptr == '\0';
}

bool solveInputString(char *input, double *num1, double *num2, char *op)
{
    char num1_str[50], num2_str[50];
    int count = sscanf(input, "%49s %c %49s", num1_str, op, num2_str);

    if (count != 3 || strlen(num1_str) == 0 || strlen(num2_str) == 0 || *op == '\0') {
        return false;
    }

    if ((!isNumeric(num1_str) && strcmp(num1_str, "ANS") != 0) ||
        (!isNumeric(num2_str) && strcmp(num2_str, "ANS") != 0)) {
        return false;
    }

    if (strcmp(num1_str, "ANS") == 0)
        *num1 = getAns();
    else
        *num1 = atof(num1_str);

    if (strcmp(num2_str, "ANS") == 0)
        *num2 = getAns();
    else
        *num2 = atof(num2_str);

    return true;
}
```

```

nguyentaiz10@TANTAIZ10H:/mnt/d/Workspace/HOH232/LAB1/Final Lab 1/2212998$ cat logic.c
#include "process.h"

double calculationLogic(double *num1, double *num2, char *op)
{
    double ans = 0;
    switch (*op)
    {
        case '+':
            ans = addition(num1, num2);
            break;
        case '-':
            ans = subtraction(num1, num2);
            break;
        case '*':
            ans = multiplication(num1, num2);
            break;
        case '/':
            ans = division(num1, num2);
            break;
        case '%':
            ans = modulo(num1, num2);
            break;
        default:
            break;
    }
    return ans;
}

double addition(double *num1, double *num2)
{
    return *num1 + *num2;
}

double subtraction(double *num1, double *num2)
{
    return *num1 - *num2;
}

double multiplication(double *num1, double *num2)
{
    return *num1 * *num2;
}

double division(double *num1, double *num2)
{
    // Do not need to check here one more time
    return *num1 / *num2;
}

double modulo(double *num1, double *num2)
{
    // Do not need to check here one more time
    return (int)(*num1) % (int)(*num2);
}

```

- Folder build: là nơi chứa các file khi make sẽ được tạo ra, và build/bin/calc là file được dùng để chạy chương trình
- Cách chạy chương trình
 - + Cách 1: make run
 - + Cách 2: make all ➔ ./ build/bin/calc
- Cách xóa các file được make ra: make clean