

อธิบายการออกแบบโปรแกรม วิธีการจัดการแต่ละส่วน

ภาพรวมของโปรแกรม โดยโปรแกรมนี้เป็นโปรแกรมจำลองการทำงานของ Vending machine โดยมี Supplier (มีหน้าที่เติมสินค้า) จำนวน 5 ราย ซึ่งเป็นตัวแทนจำหน่ายสินค้า 5 ชนิด และ Consumer (มีหน้าที่ซื้อสินค้า) จำนวน 8 คน

โดยแต่ละ Supplier และ Consumer จะกระทำกับสินค้าได้เพียง 1 ชนิดและ 1 ชิ้นต่อครั้งเท่านั้น และจำนวนสินค้าภายในตู้ มีจำนวนได้สูงสุดเท่ากับ 100 ชิ้นต่อชนิด และมีเงื่อนไขว่า Supplier และ Consumer ของสินค้าแต่ละชนิด ห้ามกระทำกับสินค้าพร้อมกัน กล่าวคือ หาก Supplier กำลังเติมสินค้าอยู่ Consumer ห้ามซื้อสินค้าชนิดนั้น หรือ หาก Consumer กำลังซื้อสินค้าอยู่ Supplier ห้ามเติมสินค้าเช่นกัน

เมื่อสินค้าที่ Supplier ต้องการเติมมีจำนวนเท่ากับ 100 (สินค้าเต็ม) Supplier จะต้องรอ เป็นระยะเวลาหนึ่ง (interval time) จึงจะสามารถกลับมาเติมสินค้าชนิดนั้นได้อีก ทำเช่นนั้นจนกว่าสินค้ามีจำนวนน้อยกว่า 100 จึงจะสามารถเติมสินค้าชนิดนั้นได้ และหาก Supplier ไม่สามารถเติมสินค้าได้ติดต่อกันเท่ากับค่า repeat จะส่งผลให้ค่า interval time เพิ่มขึ้นเป็น 2 เท่าจากค่าเดิม และสูงสุดที่ 60 วินาที

เมื่อสินค้าที่ Consumer ต้องการซื้อ มีจำนวนเท่ากับ 0 (สินค้าหมด) Consumer จะต้องรอ เป็นระยะเวลาหนึ่ง (interval time) จึงจะสามารถกลับมาซื้อสินค้าชนิดนั้นได้อีก ทำเช่นนั้นจนกว่าสินค้ามีจำนวนมากกว่า 0 จึงจะสามารถซื้อสินค้าชนิดนั้นได้ และหาก Consumer ไม่สามารถซื้อสินค้าได้ติดต่อกันเท่ากับค่า repeat จะส่งผลให้ค่า interval time เพิ่มขึ้นเป็น 2 เท่าจากค่าเดิม และสูงสุดที่ 60 วินาที

วิธีการจัดการแต่ละส่วน

ส่วนที่ 1 สร้าง Suppliers และ Consumers : โปรแกรมนี้จะทำการสร้าง thread ขึ้นมาเพื่อเป็นตัวแทนของแต่ละ Supplier และ Consumer โดยจะเก็บรายละเอียดทั้งหมดของ Supplier และ Consumer ไว้ใน Configuration files เมื่อโปรแกรมเริ่มทำงานจะทำการอ่านข้อมูลจากfileเหล่านั้น โดยแต่ละ Supplier และ Consumer จะมีรายละเอียดดังนี้

1. ชื่อสินค้าที่จะเติม หรือ ขาย
2. ค่า interval time
3. ค่า repeat

```
//Read information from supplier configuration file
void config_sup(int num){
    FILE* fp;
    char file_name[50];
    for(int i = 1; i <= num; i++){
        sprintf(file_name, "supplier%d.txt", i);
        fp = fopen(file_name, "r");

        fscanf(fp, "%[^\\n\\t]", supplier_names[i-1]);
        fscanf(fp, "%d", &supplier_interval[i-1]);
        fscanf(fp, "%d", &supplier_repeat[i-1]);
        fclose(fp);
    }
}

//Read information from consumer configuration file
void config_con(int num){
    FILE* fp;
    char file_name[50];
    for(int i = 1; i <= num; i++){
        sprintf(file_name, "consumer%d.txt", i);
        fp = fopen(file_name, "r");

        fscanf(fp, "%[^\\n\\t]", consumer_names[i-1]);
        fscanf(fp, "%d", &consumer_interval[i-1]);
        fscanf(fp, "%d", &consumer_repeat[i-1]);
        fclose(fp);
    }
}
```

```

int main(void)
{
    srand(time(NULL));

    //Read information from 5 supplier configuration files
    config_sup(SUPPLIER_NUM);
    //Read information from 8 consumer configuration files
    config_con(CONSUMER_NUM);

    //Match suppliers and consumers to item number
    for(int i = 0; i < SUPPLIER_NUM; i++){
        supplier_num[i] = i;
    }
    for(int i = 0; i < CONSUMER_NUM; i++){
        consumer_num[i] = i;
        for(int j = 0; j < SUPPLIER_NUM; j++) {
            if(!strcmp(consumer_names[i], supplier_names[j])) {
                consumedItem_num[i] = j;
            }
        }
    }

    int err;

    //Create supplier threads
    for(int i = 0; i < SUPPLIER_NUM; i++)
    {
        err = pthread_create(&(suppliers_t[i]), NULL, &addUnits, &supplier_num[i]);
        if (err != 0)
            printf("\ncan't create thread :[%s]", strerror(err));
    }

    //Create consumer threads
    for(int i = 0; i < CONSUMER_NUM; i++)
    {
        err = pthread_create(&(consumers_t[i]), NULL, &removeUnits, &consumer_num[i]);
        if (err != 0)
            printf("\ncan't create thread :[%s]", strerror(err));
    }

    pthread_exit(NULL);
}

```

ส่วนที่ 2 เติมสินค้า(เพิ่มจำนวนสินค้า) : โปรแกรมนี้จะอนุญาตให้มี Supplier หรือ Consumer เพียง 1 คน ที่สามารถเติม หรือ ซื้อสินค้านิดเดียวกัน ได้ในเวลานั้น (1ชิ้น ต่อครั้ง) โดย Supplier และ Consumer อื่นๆจะต้องรอให้ Supplier หรือ Consumer คนก่อนหน้า ทำงานเสร็จก่อน

เมื่อสินค้านิดใดชนิดหนึ่ง มีจำนวนเท่ากับ 100 Supplierที่เกี่ยวข้องจะไม่สามารถเติมสินค้าได้ และ จะต้องรอเป็นระยะเวลาหนึ่ง (interval time) จึงจะสามารถกลับมาเติมสินค้านิดนั้นได้อีก

ชื่อฟังก์ชัน addUnits

```
//Add item into vending machine
void* addUnits(void *arg)
{
    unsigned long i = 0;
    //num is supplier number
    int num = *((int*)arg);
    int count_wait = 0; //for repeat
    int interval = supplier_interval[num];

    //Time stamp
    time_t rawtime;
    struct tm * timeinfo;

    //Check if interval is over 60
    if(interval > 60) {
        interval = 60;
    }

    while(1) {
        //Permit only one thread to do work once a time
        pthread_mutex_lock(&mutex[num]);

        char *output = malloc(100);

        time ( &rawtime );
        timeinfo = localtime ( &rawtime );

        //Print out time and date
        sprintf(output, "%s %s %d %02d:%02d:%02d %d", day_name[timeinfo->tm_wday], month_name[timeinfo->tm_mon], timeinfo->tm_mday, timeinfo->tm_hour, timeinfo->tm_min, timeinfo->tm_sec, timeinfo->tm_year+1900);

        //If the item is not full, so supplier can add an item
        if(item_counters[num] < 100) {
            printf("\033[0m%s \033[1;32m%s supplied 1 unit. Stock after = %d\n", output, supplier_names[num], interval);
            item_counters[num]++;

            interval = supplier_interval[num];
            if(interval > 60) {
                interval = 60;
            }

            count_wait = 0;

            //printf("Remaining items %s: %d\n\n", supplier_names[num], item_counters[num]);
        }

        //Otherwise supplier has to wait for interval time
        else {
            printf("\033[0;32m");
            count_wait++;

            //If count_wait is equal to repeat value then double up the interval value
            if(count_wait >= supplier_repeat[num]){
                interval = interval*2;
            }
        }
    }
}
```

ส่วนที่ 3 ซื้อสินค้า(ลดจำนวนสินค้า) : โปรแกรมนี้จะอนุญาตให้มี Supplier หรือ Consumer เพียง 1 คน ที่สามารถเติม หรือ ซื้อสินค้าชนิดเดียวกันได้ในเวลานั้น (1ชิ้นต่อครั้ง) โดย Supplier และ Consumer อื่นๆจะต้องรอให้ Supplier หรือ Consumer คนก่อนหน้า ทำงานเสร็จก่อน

เมื่อสินค้าชนิดใดชนิดหนึ่ง มีจำนวนเท่ากับ 0 Consumer ที่เกี่ยวข้องจะไม่สามารถซื้อสินค้าได้ และ จะต้องรอเป็นระยะเวลาหนึ่ง (interval time) จึงจะสามารถกลับมาซื้อสินค้าชนิดนั้นได้อีก

ข้อฟังก์ชัน removeUnits

```
//Remove item from vending machine
void* removeUnits(void *arg)
{
    unsigned long i = 0;
    //num is consumer number
    int num = *(int*)arg;
    int count_wait = 0; //for repeat
    int interval = consumer_interval[num];

    //Time stamp
    time_t rawtime;
    struct tm * timeinfo;

    //Check if interval is over 60
    if(interval > 60) {
        interval = 60;
    }

    while(1) {
        //Permit only one thread to do work once a time
        pthread_mutex_lock( &mutex[consumedItem_num[num]] );

        char *output = malloc(100);

        time ( &rawtime );
        timeinfo = localtime ( &rawtime );

        //Print out time and date
        sprintf(output, "%s %s %d %02d:%02d:%02d %d",day_name[timeinfo->tm_wday],month_name[timeinfo->tm_mon],timeinfo->tm_mday, timeinfo->tm_hour, timeinfo->tm_min, timeinfo->tm_sec ,timeinfo->tm_year+1900);

        //If the item is not empty, so consumer can consume an item
        if(item_counters[consumedItem_num[num]] > 0) {
            printf("\033[0m%s \033[1;31m%s(%d) consumed 1 unit. stock after = %d\n",output, supplier_names[consumedItem_num[num]],num, interval);
            item_counters[consumedItem_num[num]]--;

            interval = consumer_interval[num];
            if(interval > 60) {
                interval = 60;
            }

            count_wait = 0;

            //printf("Remaining items %s: %d\n", supplier_names[consumedItem_num[num]], item_counters[consumedItem_num[num]]);
        }

        //Otherwise consumer has to wait for interval time
        else {
            count_wait++;

            //If count_wait is equal to repeat value then double up the interval value
            if(count_wait >= consumer_repeat[num]){
                interval = interval*2;

                //Check if interval is over 60
                if(interval > 60) {
                    interval = 60;
                }

                printf("\033[0m%s \033[0;31m%s(%d) consumer going to wait %d sec (%d/%d)\n",output, supplier_names[consumedItem_num[num]],num, interval, count_wait, consumer_repeat[num]);
                count_wait = 0;
            }
            else {
                printf("\033[0m%s \033[0;31m%s(%d) consumer going to wait %d sec (%d/%d)\n",output, supplier_names[consumedItem_num[num]],num, interval, count_wait, consumer_repeat[num]);
            }
        }

        //Unlock and give permission to the others
        pthread_mutex_unlock( &mutex[consumedItem_num[num]] );

        //Force consumer to wait for interval time
        sleep(interval);
        free(output);
    }

    return NULL;
}
```

ส่วนที่ 4 ส่วนแสดงผล : เมื่อสินค้าถูกเติม หรือ ซื้อ จาก Supplier หรือ Consumer โปรแกรมจะแสดงผลข้อความ ซึ่งประกอบไปด้วย วัน เดือน วันที่ เวลา(ชั่วโมง:นาที:วินาที) และ ปีค.ศ ตามลำดับ

จากนั้นจะแสดงผลข้อความว่า สินค้าชนิดใดถูกเติม หรือ ถูกซื้อไป และSupplier หรือ Consumer นั้นจะต้องรอเป็นระยะเวลาเท่าใด จึงจะสามารถกลับมาเติม หรือ ซื้อสินค้าชนิดนั้นได้อีก

แสดงผล Supplier

```
//Print out time and date
sprintf(output, "%s %s %d %02d:%02d:%02d %d", day_name[timeinfo->tm_wday], month_name[timeinfo->tm_mon], timeinfo->tm_mday, timeinfo->tm_hour, timeinfo->tm_min, timeinfo->tm_sec, timeinfo->tm_year+1900);
printf("\033[0m%32m%32m supplied 1 unit. Stock after = %d\n", output, supplier_names[num], interval);
printf("\033[0m%32m%32m supplier going to wait %d sec (%d/%d)\n", output, supplier_names[num], interval, count_wait, supplier_repeat[num]);
```

แสดงผล Consumer

```
//Print out time and date
sprintf(output, "%s %s %d %02d:%02d:%02d %d", day_name[timeinfo->tm_wday], month_name[timeinfo->tm_mon], timeinfo->tm_mday, timeinfo->tm_hour, timeinfo->tm_min, timeinfo->tm_sec, timeinfo->tm_year+1900);
printf("\033[0m%32m%32m consumed 1 unit. stock after = %d\n", output, supplier_names[consumedItem_num[num]], num, interval);
printf("\033[0m%32m%32m consumer going to wait %d sec (%d/%d)\n", output, supplier_names[consumedItem_num[num]], num, interval, count_wait, consumer_repeat[num]);
```

กำหนดแต่ละ Consumer ดังนี้ (ไฟล์ .txt)

Consumer 1

Pepsi
interval time = 6
repeat time = 2

Consumer 2

Coca-Cola
interval time = 4
repeat time = 2

Consumer 3

Mountain Dew
interval time = 8
repeat time = 3

Consumer 4

Starbucks
interval time = 12
repeat time = 3

Consumer 5

Singha
interval time = 2
repeat time = 4

Consumer 6

Pepsi
interval time = 18
repeat time = 4

Consumer 7

Mountain Dew
interval time = 30
repeat time = 5

Consumer 8

Singha
interval time = 14
repeat time = 5

กำหนดแต่ละ Supplier ดังนี้ (ไฟล์ .txt)

Supplier 1

Pepsi

interval time = 1

repeat time = 2

Supplier 2

Coca-Cola

interval time = 2

repeat time = 3

Supplier 3

Mountain Dew

interval time = 3

repeat time = 4

Supplier 4

Starbucks

interval time = 5

repeat time = 5

Supplier 5

Singha

interval time = 7

repeat time = 6

ผลการรันของโปรแกรม

```
vending-machine — -bash — 119x50
[Sutarats-Air:vending-machine ssuzy$ gcc -pthread vm.c
[Sutarats-Air:vending-machine ssuzy$ ./a.out
Thu Apr 19 15:32:46 2018 Starbucks supplied 1 unit. Going to wait for 5 sec
Remaining items Starbucks: 1

Thu Apr 19 15:32:46 2018 Coca-Cola supplied 1 unit. Going to wait for 2 sec
Remaining items Coca-Cola: 1

Thu Apr 19 15:32:46 2018 Singha supplied 1 unit. Going to wait for 7 sec
Remaining items Singha: 1

Thu Apr 19 15:32:46 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 1

Thu Apr 19 15:32:46 2018 Coca-Cola(Thread 1) consumed 1 unit. Going to wait for 4 sec
Remaining items Coca-Cola: 0

Thu Apr 19 15:32:46 2018 Mountain Dew supplied 1 unit. Going to wait for 3 sec
Remaining items Mountain Dew: 1

Thu Apr 19 15:32:46 2018 Mountain Dew(Thread 2) consumed 1 unit. Going to wait for 8 sec
Remaining items Mountain Dew: 0

Thu Apr 19 15:32:46 2018 Starbucks(Thread 3) consumed 1 unit. Going to wait for 12 sec
Remaining items Starbucks: 0

Thu Apr 19 15:32:46 2018 Singha(Thread 4) consumed 1 unit. Going to wait for 2 sec
Remaining items Singha: 0

Thu Apr 19 15:32:46 2018 Mountain Dew is out of order. Mountain Dew(Thread 6) consumer going to wait for 30 sec (1/5)

Thu Apr 19 15:32:46 2018 Singha is out of order. Singha(Thread 7) consumer going to wait for 14 sec (1/5)

Thu Apr 19 15:32:46 2018 Pepsi(Thread 0) consumed 1 unit. Going to wait for 6 sec
Remaining items Pepsi: 0

Thu Apr 19 15:32:46 2018 Pepsi is out of order. Pepsi(Thread 5) consumer going to wait for 18 sec (1/4)

Thu Apr 19 15:32:47 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 1

Thu Apr 19 15:32:48 2018 Coca-Cola supplied 1 unit. Going to wait for 2 sec
Remaining items Coca-Cola: 1

Thu Apr 19 15:32:48 2018 Singha is out of order. Singha(Thread 4) consumer going to wait for 2 sec (1/4)

Thu Apr 19 15:32:48 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 2

Thu Apr 19 15:32:49 2018 Mountain Dew supplied 1 unit. Going to wait for 3 sec
```



```
Thu Apr 19 15:32:48 2018 Singha is out of order. Singha(Thread 4) consumer going to wait for 2 sec (1/4)

Thu Apr 19 15:32:48 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 2

Thu Apr 19 15:32:49 2018 Mountain Dew supplied 1 unit. Going to wait for 3 sec
Remaining items Mountain Dew: 1

Thu Apr 19 15:32:49 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 3

Thu Apr 19 15:32:50 2018 Coca-Cola(Thread 1) consumed 1 unit. Going to wait for 4 sec
Remaining items Coca-Cola: 0

Thu Apr 19 15:32:50 2018 Coca-Cola supplied 1 unit. Going to wait for 2 sec
Remaining items Coca-Cola: 1

Thu Apr 19 15:32:50 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 4

Thu Apr 19 15:32:50 2018 Singha is out of order. Singha(Thread 4) consumer going to wait for 2 sec (2/4)

Thu Apr 19 15:32:51 2018 Starbucks supplied 1 unit. Going to wait for 5 sec
Remaining items Starbucks: 1

Thu Apr 19 15:32:51 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 5

Thu Apr 19 15:32:52 2018 Pepsi(Thread 0) consumed 1 unit. Going to wait for 6 sec
Remaining items Pepsi: 4

Thu Apr 19 15:32:52 2018 Mountain Dew supplied 1 unit. Going to wait for 3 sec
Remaining items Mountain Dew: 2

Thu Apr 19 15:32:52 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 5

Thu Apr 19 15:32:52 2018 Singha is out of order. Singha(Thread 4) consumer going to wait for 2 sec (3/4)

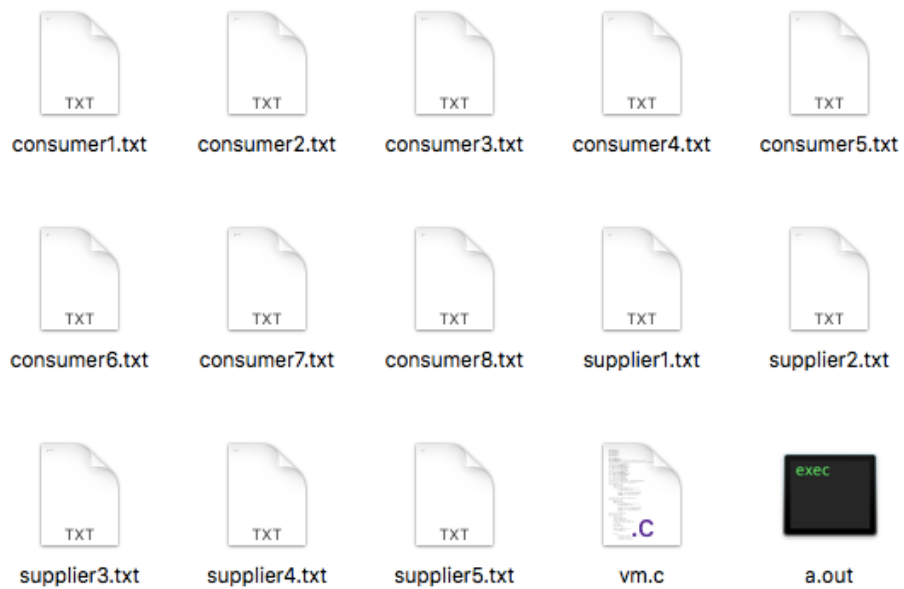
Thu Apr 19 15:32:52 2018 Coca-Cola supplied 1 unit. Going to wait for 2 sec
Remaining items Coca-Cola: 2

Thu Apr 19 15:32:53 2018 Singha supplied 1 unit. Going to wait for 7 sec
Remaining items Singha: 1

Thu Apr 19 15:32:53 2018 Pepsi supplied 1 unit. Going to wait for 1 sec
Remaining items Pepsi: 6

Thu Apr 19 15:32:54 2018 Mountain Dew(Thread 2) consumed 1 unit. Going to wait for 8 sec
```

ไฟล์ที่ใช้ทั้งหมด



รายชื่อสมาชิกกลุ่ม

นายวิชยุตม์ โลหะอุดม	รหัสนักศึกษา 5801150
นางสาวสุรารัตน์ สุกธิธาทิพย์	รหัสนักศึกษา 58011348
นายอรรถ อัสโสรัตน์กุล	รหัสนักศึกษา 58011424