

# 作品集







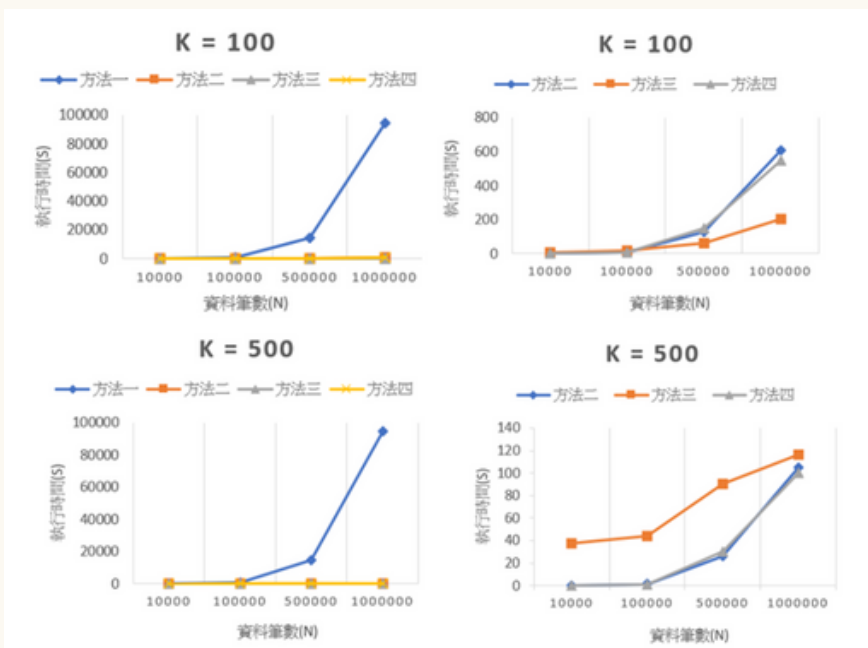
## Compiler

程式語言 C / C++

課程 系統程式

說明 實作含指定statement的Compiler，依各指令放入相對應的Table中並轉換為中間碼，最後將中間碼及所對應的資訊輸出。

Line	Location	Source code	Object code
1	0000	COPY START 0	
2	0001	var1 RESW 1	
3	0002	var2 RESB 12	
4	0003	var3 EQU 12	
5	0004	add var1	
6	0005	+addf var2	1B2FEE
7	0006	addr a,b	5B100003
8	0007	+and var1	9003
9	0008	clear b	43100000
10	0009	comp 0	B430
11	0010	+compf var3	2B0000
12	0011	compr b,a	8B10000C
13	0012	.this is comment	A030
14	0013	div var1	272FD6
15	0014	divf var1	672FD3
16	0015	divr l,s	9C24
17	0016	fix	C4
18	0017	float	C0
19	0018	END FIRST	



## MultiProcess與MultiThread 排序時間比較

程式語言 Python

課程 作業系統

說明 使用MultiProcess及MultiThread分別執行BubbleSort及MergeSort，比較分割成不同個數的Process及Thread在不同資料量(1、10、50、100萬)執行時間的差異。

	FCFS	RR	SJF	SRTF	HRRN	PPRR
input1	14.33	18.40	8.87	8.07	11.60	14.67
Input2	8.40	6.40	8.20	3	8.20	9.40
Input3	6.67	11.67	6.67	6.67	6.67	12.50
Input4	3.75	5.50	3.50	3.25	3.75	4.50

表 1：不同 input 檔個別使用六種排程法的平均等待時間(ms)

	FCFS	RR	SJF	SRTF	HRRN	PPRR
input1	18.20	22.27	12.73	11.93	15.47	18.53
Input2	13.20	11.20	13	7.80	13	14.20
Input3	24.17	29.17	24.17	24.17	24.17	30
Input4	8.75	10.50	8.50	8.25	8.75	9.50

表 2：不同 input 檔個別使用六種排程法的平均往返時間(ms)

## CPU Scheduling

程式語言 Python

課程 作業系統

說明 模擬不同CPU排程演算法，包含FCFS(先進先出)、RR(知更鳥是循環)、SJF(最短工作優先)、SRTF(最短剩餘時間優先)、HRRN(最高響應比率優先)、PPRR(優先等級+RR)，並比較不同排程法間差異。

## Paging Management

程式語言 Python

課程 作業系統

說明 實作Memory中的分頁法，包括FIFO(先進先出)、LRU(最近罕用)、LFU+FIFO、MFU+FIFO、LFU+LRU。

Page frame number 皆為 3

Input1 Reference String : 123412512345

Input2 Reference String : 70120304230321201701

	FIFO	LRU	LFU+FIFO	MFU+FIFO	LFU+LRU
Input1	9	10	10	9	10
Input2	15	12	13	15	11

表 1：Page Fault 次數比較表

	FIFO	LRU	LFU+FIFO	MFU+FIFO	LFU+LRU
Input1	6	7	7	6	7
Input2	12	9	10	12	8

表 2：Page Replacement 次數比較表





## 推箱子遊戲

程式語言 Python

課程 演算法分析

說明 S為人初始位置，B為箱子之初始位置，T為目標位置，依輸入之迷宮，將箱子推動至指定目標，E、W、S、N、e、w、s、n分別代表推動或移動方向。

```
import time
from queue import Queue

r, c = 0, 0

maze = [[]]

d = [[-1, 1, 0, 0], [0, 0, -1, 1]]

push = ['N', 'S', 'W', 'E']
move = ['n', 's', 'w', 'e']

tmp = ""

class P:
    def __init__(self, x, y, px, py, ans):
        self.x = x
        self.y = y
        self.px = px
        self.py = py
        self.ans = ans

def valid(x, y):
    if (x >= 0 and x < r and y >= 0 and y < c):
        if maze[x][y] != 'x': return True
    else: return False
    else: return False

Anaconda Powershell Prompt
請輸入迷宮的列數與行數: 7 11
請輸入每一列的迷宮: #####
請輸入每一列的迷宮: #T##.....#
請輸入每一列的迷宮: #.#.#.####
請輸入每一列的迷宮: #...B...#
請輸入每一列的迷宮: #.....S...#
請輸入每一列的迷宮: #####
Maze #1
eeennwwwwweeeeeeesswwwwwnnNN
總共花費 0.00786280632019043 秒
請輸入迷宮的列數與行數: |
```

## 以關鍵字建2-3樹及AVL樹

程式語言 C / C++

課程 資料結構

說明 將學校代碼/學校名稱/科系代碼/科系名稱/日間/進修別/等級別/學生數/教師數/上學年度畢業生數/縣市名稱/體育別等資訊分別按照學校名稱建 2-3 tree；按照上學年度畢業生數建 AVL tree。

```
*** Search Tree Utilities ***
* 0. Quit
* 1. Build a 2-3tree
* 2. Build a AVLtree
*****
Please input command(0, 1, 2): 1

Input a file number ([0]Quit): 203
Tree height: 4
1: [56] 國立成功大學, 法律學系, 0 日, 0 學士, 205
2: [57] 國立成功大學, 生命科學系, 0 日, 0 博士, 24
3: [58] 國立成功大學, 生命科學系, 0 日, 0 博士, 62
4: [59] 國立成功大學, 生命科學系, 0 日, 0 博士, 196
5: [60] 國立成功大學, 熱帶植物科學研究所, 0 日, 0 博士, 19
6: [61] 國立成功大學, 海洋科技與管理研究所, 0 日, 0 博士, 13
7: [62] 國立成功大學, 海洋科技與管理研究所, 0 日, 0 博士, 18
8: [63] 國立成功大學, 自然資源與環境管理學系, 0 日, 0 博士, 14
9: [64] 國立成功大學, 數學系, 0 日, 0 博士, 59
10: [65] 國立成功大學, 數學系, 0 日, 0 博士, 200
11: [66] 國立成功大學, 數學系, 0 日, 0 博士, 461
12: [67] 國立成功大學, 系統及船舶機械工程學系, 0 日, 0 博士, 205
13: [68] 國立成功大學, 土木工程學系, 0 日, 0 博士, 39
14: [69] 國立成功大學, 土木工程學系, 0 日, 0 博士, 235
15: [70] 國立成功大學, 土木工程學系, 0 日, 0 博士, 235
16: [71] 國立成功大學, 土木工程學系, 0 日, 0 博士, 456
17: [72] 國立成功大學, 海洋工程學系, 0 日, 0 博士, 219
18: [73] 國立成功大學, 公共衛生與預防醫學系, 0 日, 0 博士, 28
19: [74] 國立成功大學, 海運及空運管理學系, 0 日, 0 博士, 24
20: [75] 國立成功大學, 海運及空運管理學系, 0 日, 0 博士, 59
21: [76] 國立成功大學, 海運及空運管理學系, 0 日, 0 博士, 210
22: [77] 國立成功大學, 機械工程學系, 0 日, 0 博士, 27
23: [78] 國立成功大學, 機械工程學系, 0 日, 0 博士, 97
24: [79] 國立成功大學, 護理學系, 0 日, 0 博士, 224
25: [80] 國立成功大學, 護理學系, 0 日, 0 博士, 146

*** Search Tree Utilities ***
* 0. Quit
* 1. Build a 2-3tree
* 2. Build a AVLtree
*****
Please input command(0, 1, 2): 2
Tree height: 0
1: [12] 國立臺灣大學, 政治學系, 0 日, 0 學士, 761
2: [190] 國立中山大學, 政治學系, 0 日, 0 博士, 21
3: [190] 國立中山大學, 政治學系, 0 日, 0 博士, 56
4: [197] 國立中山大學, 政治學系, 0 日, 0 博士, 84
5: [543] 中國文化大學, 政治學系, 0 日, 0 博士, 44
6: [544] 中國文化大學, 政治學系, 0 日, 0 博士, 24
```

## 自動販賣機

程式語言 Verilog

課程 邏輯設計實驗

說明 利用有限狀態機的方法模擬出自動販賣機。共分為四個狀態：初始狀態(S0)、選擇狀態(S1)、給予狀態(S2)、結帳狀態(S3)，若購買完成或取消購買則回到初始狀態。

```
1 module AutoVendor( clk, reset, coin, drink_choose, total_coin, refund );
2
3 input clk, reset;
4 input [6:0] coin;
5 input [2:0] drink_choose;
6 output reg [6:0] refund, total_coin;
7 reg tea, coke, coffee, milk;
8 reg [1:0] state, next_state;
9
10 parameter S0 = 2'b00;
11 parameter S1 = 2'b01;
12 parameter S2 = 2'b10;
13 parameter S3 = 2'b11;
14
15 always @(posedge clk or reset) begin
16     if (reset == 1'b1 && total_coin == 0) begin
17         if (refund == total_coin) $display("exchange 0 dollars");
18         else $display("exchange ", total_coin, " dollars");
19         state = S0;
20         tea = 1'b0;
21         coke = 1'b0;
22         coffee = 1'b0;
23         milk = 1'b0;
24         total_coin = 7'b000000;
25         refund = 7'b000000;
26     end
27 end
```