學校不會叫的版本控制

國立屏東大學
電腦科學與人工智慧學系

咪路 <mail@mirumo.org />

程式開發經驗？

專題 / 中小型專案開發

老師說：
「努力會有回報的，多寫一個功能加十分」

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

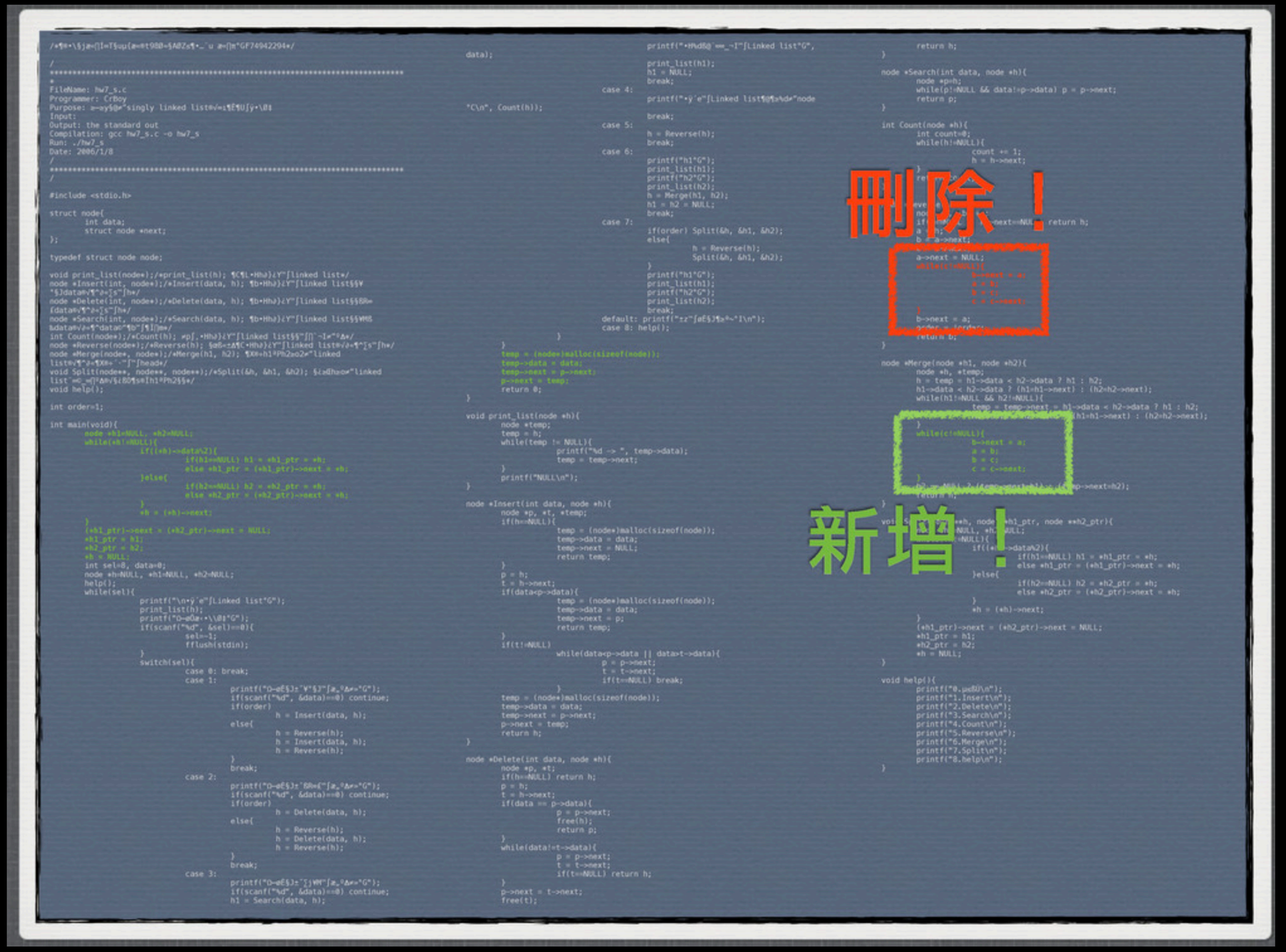
於是....你開始努力

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

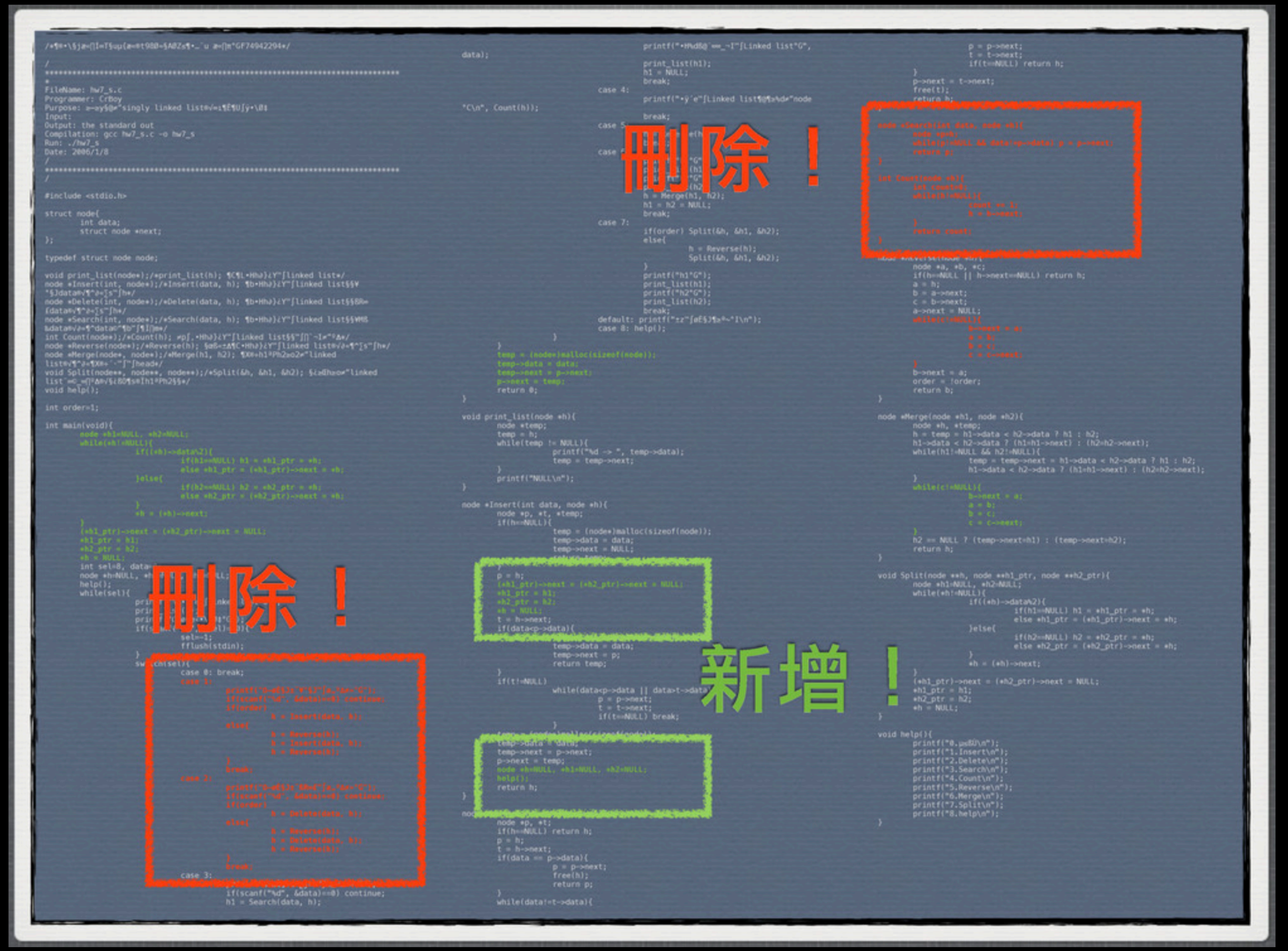
```
/*¶¤•\§ja=[]I=T§uq{a=¢t988=$ABZx¶•_`u a=(]n"GF74942294+/

/
**************************************************************************
*
FileName: hw7_s.c
Programmer: CrBoy
Purpose: a=ay$@>"singly linked list¤√w±¶£¶Ujÿ•\Ø‡
Input:
Output: the standard out
Compilation: gcc hw7_s.c -o hw7_s
Run: ./hw7_s
Date: 2006/1/8
/
**************************************************************************
/

#include <stdio.h>

struct node{
        int data;
        struct node *next;
};

typedef struct node node;

void print_list(node*);/*print_list(h); ¶C¶L•Hhø}¿Y"∫linked list*/
node *Insert(int, node*);/*Insert(data, h); ¶b•Hhø}¿Y"∫linked list§§¥
"§}data¤√¶^a=]s"∫h*/
node *Delete(int, node*);/*Delete(data, h); ¶b•Hhø}¿Y"∫linked list§§8R=
{data¤√¶^a+]s"∫h*/
node *Search(int, node*);/*Search(data, h); ¶b•Hhø}¿Y"∫linked list§§WH8
&data¤√a=¶^data¤<"¶o"∫¶∏]m*/
int Count(node*);/*Count(h); ¤p∫.•Hhø}¿Y"∫linked list§§"∫∏¨~l¤="²A*/
node *Reverse(node*);/*Reverse(h); §a8<±A¶C•Hhø}¿Y"∫linked list¤√a=¶^]s"∫h*/
node *Merge(node*, node*);/*Merge(h1, h2); ¶X¤•h1²Ph2»o2¤"linked
list¤√¶^a=¶X¤•`:"∫"∫head*/
void Split(node**, node**, node**);/*Split(&h, &h1, &h2); §¿aDhuo¤"linked
list`¤©_«]²A¤√§{8O¶$¤Ih1²Ph2§§*/
void help();

int order=1;

int main(void){
        int sel=0, data=0;
        node *h=NULL, *h1=NULL, *h2=NULL;
        help();
        while(sel){
                printf("\n•ÿ´e"∫Linked list"G");
                print_list(h);
                printf("O~øÕa•*\\Ø‡"G");
                if(scanf("%d", &sel)==0){
                        sel=-1;
                        fflush(stdin);
                }
                switch(sel){
                        case 0: break;
                        case 1:
                                printf("O~øÉ§J±'¥"§}"∫æ_ª&¤="G");
                                if(scanf("%d", &data)==0) continue;
                                if(order)
                                        h = Insert(data, h);
                                else{
                                        h = Reverse(h);
                                        h = Insert(data, h);
                                        h = Reverse(h);
                                }
                                break;
                        case 2:
                                printf("O~øÉ§J±'ßR=ã"∫æ_ª&¤="G");
                                if(scanf("%d", &data)==0) continue;
                                if(order)
                                        h = Delete(data, h);
                                else{
                                        h = Reverse(h);
                                        h = Delete(data, h);
                                        h = Reverse(h);
                                }
                                break;
                        case 3:
                                printf("O~øÉ§J±'§jWH"∫æ_ª&¤="G");
                                if(scanf("%d", &data)==0) continue;
                                h1 = Search(data, h);
                                printf("•HhdØ@`==_¬I"∫Linked list"G",
data);

                                print_list(h1);
                                h1 = NULL;
                                break;
                        case 4:
                                printf("•ÿ´e"∫Linked list¶@¶¤%d="node
°C\n", Count(h));

                                break;
                        case 5:
                                h = Reverse(h);
                                break;
                        case 6:
                                printf("h1"G");
                                print_list(h1);
                                printf("h2"G");
                                print_list(h2);
                                h = Merge(h1, h2);
                                h1 = h2 = NULL;
                                break;
                        case 7:
                                if(order) Split(&h, &h1, &h2);
                                else{
                                        h = Reverse(h);
                                        Split(&h, &h1, &h2);
                                }
                                printf("h1"G");
                                print_list(h1);
                                printf("h2"G");
                                print_list(h2);
                                break;
                        default: printf("±z"∫øÉ§J¶¤ª~"1\n");
                        case 8: help();
                }
        }
        return 0;
}

void print_list(node *h){
        node *temp;
        temp = h;
        while(temp != NULL){
                printf("%d -> ", temp->data);
                temp = temp->next;
        }
        printf("NULL\n");
}

node *Insert(int data, node *h){
        node *p, *t, *temp;
        if(h==NULL){
                temp = (node*)malloc(sizeof(node));
                temp->data = data;
                temp->next = NULL;
                return temp;
        }
        p = h;
        t = h->next;
        if(data<p->data){
                temp = (node*)malloc(sizeof(node));
                temp->data = data;
                temp->next = p;
                return temp;
        }
        if(t!=NULL)
                while(data<p->data || data>t->data){
                        p = p->next;
                        t = t->next;
                        if(t==NULL) break;
                }
        temp = (node*)malloc(sizeof(node));
        temp->data = data;
        temp->next = p->next;
        p->next = temp;
        return h;
}

node *Delete(int data, node *h){
        node *p, *t;
        if(h==NULL) return h;
        p = h;
        t = h->next;
        if(data == p->data){
                p = p->next;
                free(h);
                return p;
        }
        while(data!=t->data){
                p = p->next;
                t = t->next;
                if(t==NULL) return h;
        }
        p->next = t->next;
        free(t);
        return h;
}

node *Search(int data, node *h){
        node *p=h;
        while(p!=NULL && data!=p->data) p = p->next;
        return p;
}

int Count(node *h){
        int count=0;
        while(h!=NULL){
                count += 1;
                h = h->next;
        }
        return count;
}

node *Reverse(node *h){
        node *a, *b, *c;
        if(h==NULL || h->next==NULL) return h;
        a = h;
        b = a->next;
        c = b->next;
        a->next = NULL;
        while(c!=NULL){
                b->next = a;
                a = b;
                b = c;
                c = c->next;
        }
        b->next = a;
        order = !order;
        return b;
}

node *Merge(node *h1, node *h2){
        node *h, *temp;
        h = temp = h1->data < h2->data ? h1 : h2;
        h1->data < h2->data ? (h1=h1->next) : (h2=h2->next);
        while(h1!=NULL && h2!=NULL){
                temp = temp->next = h1->data < h2->data ? h1 : h2;
                h1->data < h2->data ? (h1=h1->next) : (h2=h2->next);
        }
        h2 == NULL ? (temp->next=h1) : (temp->next=h2);
        return h;
}

void Split(node **h, node **h1_ptr, node **h2_ptr){
        node *h1=NULL, *h2=NULL;
        while(*h!=NULL){
                if((*h)->data%2){
                        if(h1==NULL) h1 = *h1_ptr = *h;
                        else *h1_ptr = (*h1_ptr)->next = *h;
                }else{
                        if(h2==NULL) h2 = *h2_ptr = *h;
                        else *h2_ptr = (*h2_ptr)->next = *h;
                }
                *h = (*h)->next;
        }
        (*h1_ptr)->next = (*h2_ptr)->next = NULL;
        *h1_ptr = h1;
        *h2_ptr = h2;
        *h = NULL;
}

void help(){
        printf("0.µø&U\n");
        printf("1.Insert\n");
        printf("2.Delete\n");
        printf("3.Search\n");
        printf("4.Count\n");
        printf("5.Reverse\n");
        printf("6.Merge\n");
        printf("7.Split\n");
        printf("8.help\n");
}
```

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

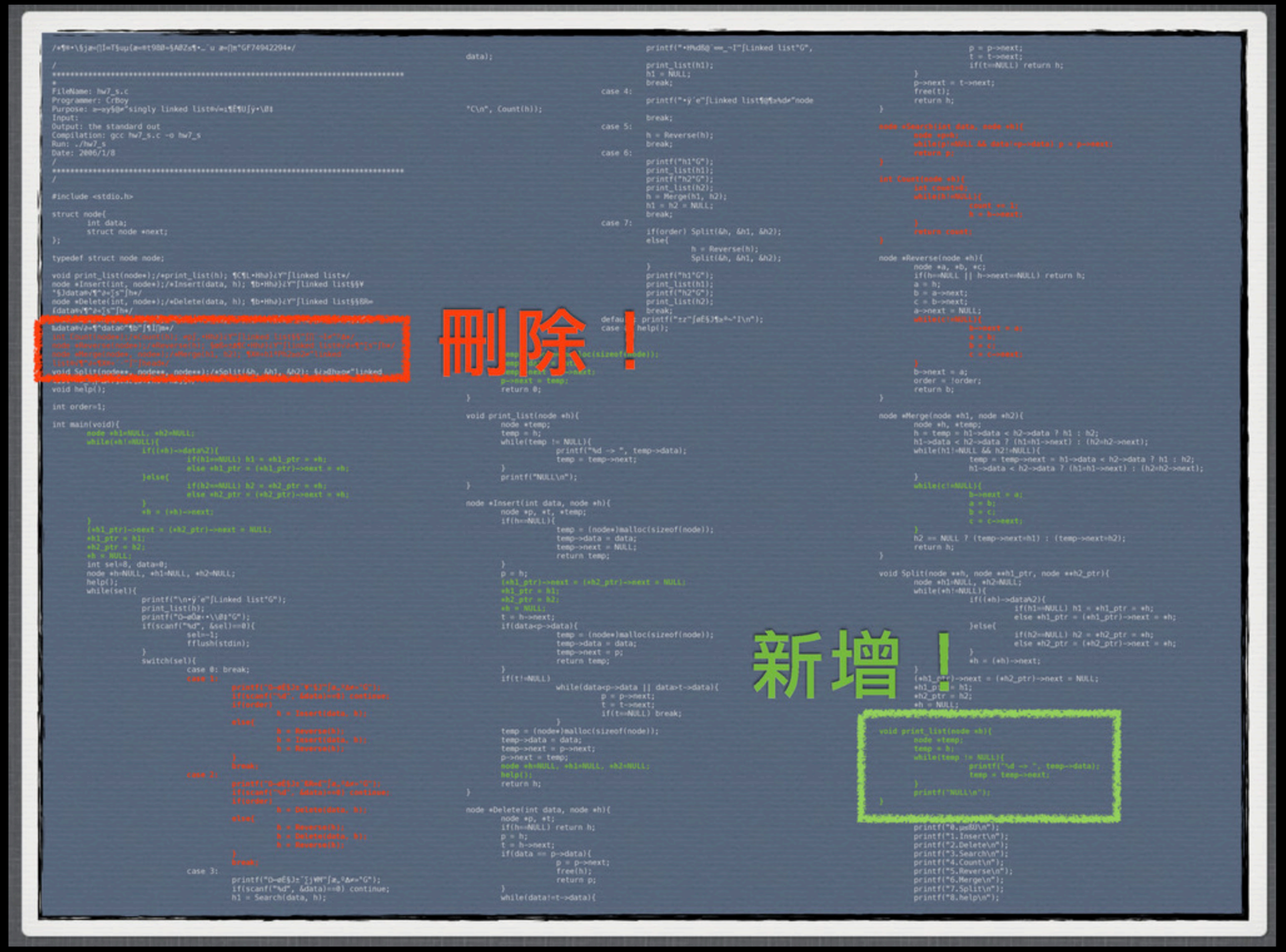
改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>



改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>



改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>



改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

你努力了兩天之後發現...

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

功能寫壞了...於是...

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

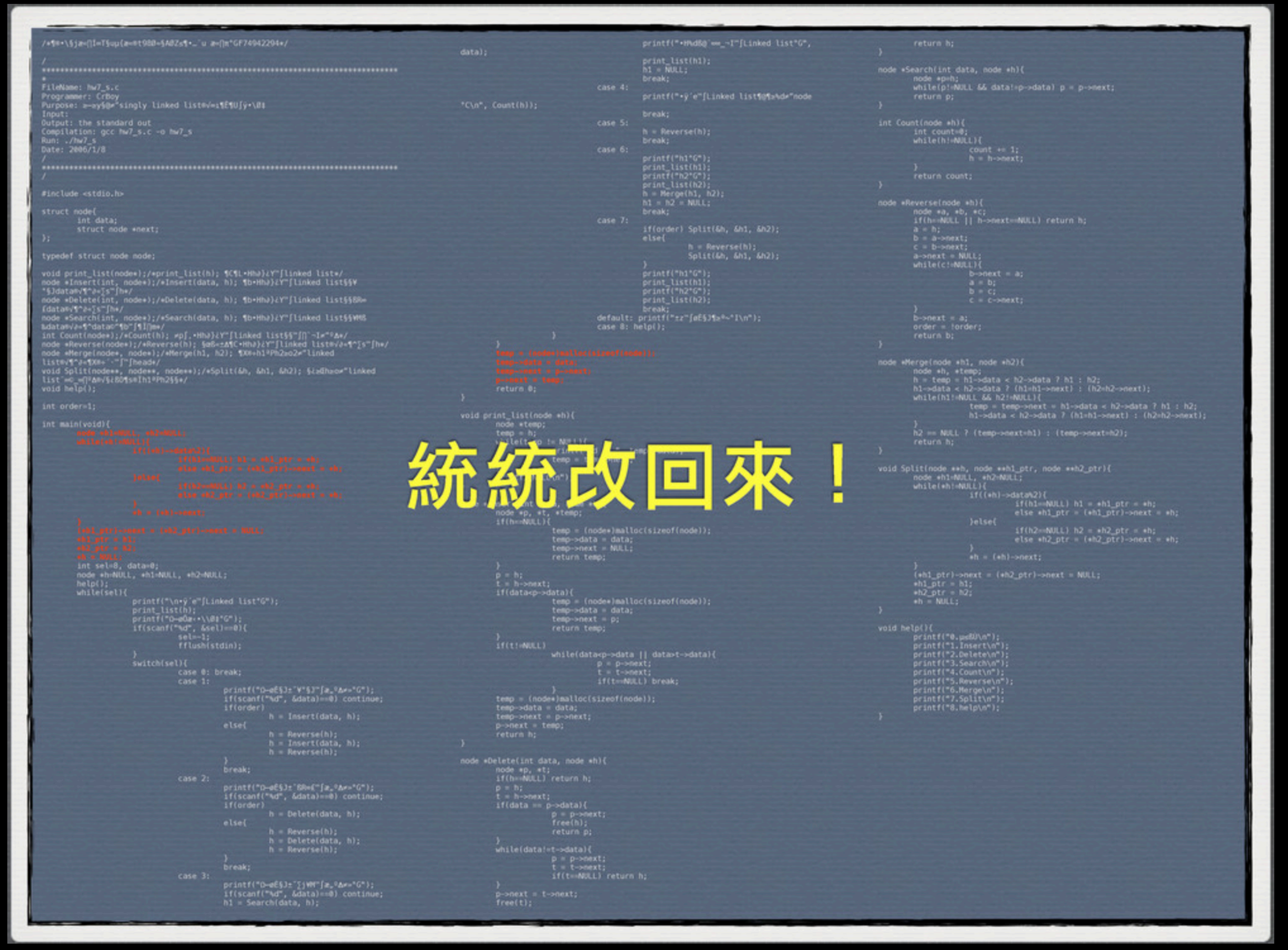
```c
/*¶®•\§jæ=ƒ]I=T§uµ{æ=®t98Ø=§AØZs¶•…´u æ=ƒ]m"GF74942294*/

/
********************************************************************************
*
FileName: hw7_s.c
Programmer: CrBoy
Purpose: ≥¬ay§@≠"singly linked list®√=ı¶Ê¶UƒŸ•\Ø‡
Input:
Output: the standard out
Compilation: gcc hw7_s.c -o hw7_s
Run: ./hw7_s
Date: 2006/1/8
/
********************************************************************************
/

#include <stdio.h>

struct node{
	int data;
	struct node *next;
};

typedef struct node node;

void print_list(node*);/*print_list(h); ¶C¶L•Hh∂}¿Y"∫linked list*/
node *Insert(int, node*);/*Insert(data, h); ¶b•Hh∂}¿Y"∫linked list§§¥
"§Jdata®√¶^∂«∑s"∫h*/
node *Delete(int, node*);/*Delete(data, h); ¶b•Hh∂}¿Y"∫linked list§§ßR∞
£data®√¶^∂«∑s"∫h*/
node *Search(int, node*);/*Search(data, h); ¶b•Hh∂}¿Y"∫linked list§§¥M®
‰data®√∂«¶^data©"¶b"∫¶Iƒ]m*/
int Count(node*);/*Count(h); ≠p∫,•Hh∂}¿Y"∫linked list§§"∫∏¨¬I≠"ºΔ*/
node *Reverse(node*);/*Reverse(h); §œß«±Δ¶C•Hh∂}¿Y"∫linked list®√∂«¶^∑s"∫h*/
node *Merge(node*, node*);/*Merge(h1, h2); ¶X®÷h1ªPh2≥o2≠"linked
list®√¶^∂«¶X®÷´-"∫"∫head*/
void Split(node**, node**, node**);/*Split(&h, &h1, &h2); §¿∂Ðh≥o≠"linked
list´=©_=ƒ]ºΔ®√§¿ßØ¶s®Ïh1ªPh2§§*/
void help();

int order=1;

int main(void){
	node *h1=NULL, *h2=NULL;
	while(*h!=NULL){
			if((*h)->data%2){
				if(h1==NULL) h1 = *h1_ptr = *h;
				else *h1_ptr = (*h1_ptr)->next = *h;
			}else{
				if(h2==NULL) h2 = *h2_ptr = *h;
				else *h2_ptr = (*h2_ptr)->next = *h;
			}
			*h = (*h)->next;
	}
	(*h1_ptr)->next = (*h2_ptr)->next = NULL;
	*h1_ptr = h1;
	*h2_ptr = h2;
	*h = NULL;
	int sel=8, data=0;
	node *h=NULL, *h1=NULL, *h2=NULL;
	help();
	while(sel){
			printf("\n•ÿ´e"∫Linked list"G");
			print_list(h);
			printf("Ω÷œÕæ‹•\\Ø‡"G");
			if(scanf("%d", &sel)==0){
				sel=-1;
				fflush(stdin);
			}
			switch(sel){
				case 0: break;
				case 1:
					printf("Ω÷œE§J±¨¥"§J"∫æ„ºΔ≠"G");
					if(scanf("%d", &data)==0) continue;
					if(order)
						h = Insert(data, h);
					else{
						h = Reverse(h);
						h = Insert(data, h);
						h = Reverse(h);
					}
					break;
				case 2:
					printf("Ω÷œE§J±¨ßR∞£"∫æ„ºΔ≠"G");
					if(scanf("%d", &data)==0) continue;
					if(order)
						h = Delete(data, h);
					else{
						h = Reverse(h);
						h = Delete(data, h);
						h = Reverse(h);
					}
					break;
				case 3:
					printf("Ω÷œE§J±¨∑j¥M"∫æ„ºΔ≠"G");
					if(scanf("%d", &data)==0) continue;
					h1 = Search(data, h);
					printf("•H%d∂}©l™∫_¬I"∫Linked list"G",
data);
					print_list(h1);
					h1 = NULL;
					break;
				case 4:
					printf("•ÿ´e"∫Linked list¶@¶≥%d≠"node
"C\n", Count(h));
					break;
				case 5:
					h = Reverse(h);
					break;
				case 6:
					printf("h1"G");
					print_list(h1);
					printf("h2"G");
					print_list(h2);
					h = Merge(h1, h2);
					h1 = h2 = NULL;
					break;
				case 7:
					if(order) Split(&h, &h1, &h2);
					else{
						h = Reverse(h);
						Split(&h, &h1, &h2);
					}
					printf("h1"G");
					print_list(h1);
					printf("h2"G");
					print_list(h2);
					break;
				default: printf("±z"∫øE§J¶≥ª~°I\n");
				case 8: help();
			}
	}
	temp = (node*)malloc(sizeof(node));
	temp->data = data;
	temp->next = p->next;
	p->next = temp;
	return 0;
}

void print_list(node *h){
	node *temp;
	temp = h;
	while(temp != NULL){
			printf("%d -> ", temp->data);
			temp = temp->next;
	}
	printf("NULL\n");
}

node *Insert(int data, node *h){
	node *p, *t, *temp;
	if(h==NULL){
			temp = (node*)malloc(sizeof(node));
			temp->data = data;
			temp->next = NULL;
			return temp;
	}
	p = h;
	(*h1_ptr)->next = (*h2_ptr)->next = NULL;
	*h1_ptr = h1;
	*h2_ptr = h2;
	*h = NULL;
	t = h->next;
	if(data<p->data){
			temp = (node*)malloc(sizeof(node));
			temp->data = data;
			temp->next = p;
			return temp;
	}
	if(t!=NULL)
			while(data<p->data || data>t->data){
				p = p->next;
				t = t->next;
				if(t==NULL) break;
			}
	temp = (node*)malloc(sizeof(node));
	temp->data = data;
	temp->next = p->next;
	p->next = temp;
	node *h=NULL, *h1=NULL, *h2=NULL;
	help();
	return h;
}

node *Delete(int data, node *h){
	node *p, *t;
	if(h==NULL) return h;
	p = h;
	t = h->next;
	if(data == p->data){
			p = p->next;
			free(h);
			return p;
	}
	while(data!=t->data){
			p = p->next;
			t = t->next;
			if(t==NULL) return h;
	}
	p->next = t->next;
	free(t);
	return h;
}

node *Search(int data, node *h){
	node *p=h;
	while(p!=NULL && data!=p->data) p = p->next;
	return p;
}

int Count(node *h){
	int count=0;
	while(h!=NULL){
			count += 1;
			h = h->next;
	}
	return count;
}

node *Reverse(node *h){
	node *a, *b, *c;
	if(h==NULL || h->next==NULL) return h;
	a = h;
	b = a->next;
	c = b->next;
	a->next = NULL;
	while(c!=NULL){
			b->next = a;
			a = b;
			b = c;
			c = c->next;
	}
	b->next = a;
	order = !order;
	return b;
}

node *Merge(node *h1, node *h2){
	node *h, *temp;
	h = temp = h1->data < h2->data ? h1 : h2;
	h1->data < h2->data ? (h1=h1->next) : (h2=h2->next);
	while(h1!=NULL && h2!=NULL){
			temp = temp->next = h1->data < h2->data ? h1 : h2;
			h1->data < h2->data ? (h1=h1->next) : (h2=h2->next);
	}
	while(c!=NULL){
			b->next = a;
			a = b;
			b = c;
			c = c->next;
	}
	h2 == NULL ? (temp->next=h1) : (temp->next=h2);
	return h;
}

void Split(node **h, node **h1_ptr, node **h2_ptr){
	node *h1=NULL, *h2=NULL;
	while(*h!=NULL){
			if((*h)->data%2){
				if(h1==NULL) h1 = *h1_ptr = *h;
				else *h1_ptr = (*h1_ptr)->next = *h;
			}else{
				if(h2==NULL) h2 = *h2_ptr = *h;
				else *h2_ptr = (*h2_ptr)->next = *h;
			}
			*h = (*h)->next;
	}
	(*h1_ptr)->next = (*h2_ptr)->next = NULL;
	*h1_ptr = h1;
	*h2_ptr = h2;
	*h = NULL;
}

void print_list(node *h){
	node *temp;
	temp = h;
	while(temp != NULL){
			printf("%d -> ", temp->data);
			temp = temp->next;
	}
	printf("NULL\n");
}

void help(){
	printf("0.µ≠ßÙ\n");
	printf("1.Insert\n");
	printf("2.Delete\n");
	printf("3.Search\n");
	printf("4.Count\n");
	printf("5.Reverse\n");
	printf("6.Merge\n");
	printf("7.Split\n");
	printf("8.help\n");
```

改自：**Code Smart, Don't Code hard** **BY 畢玉泉(CrBoy) <crboy@crboy.net>**

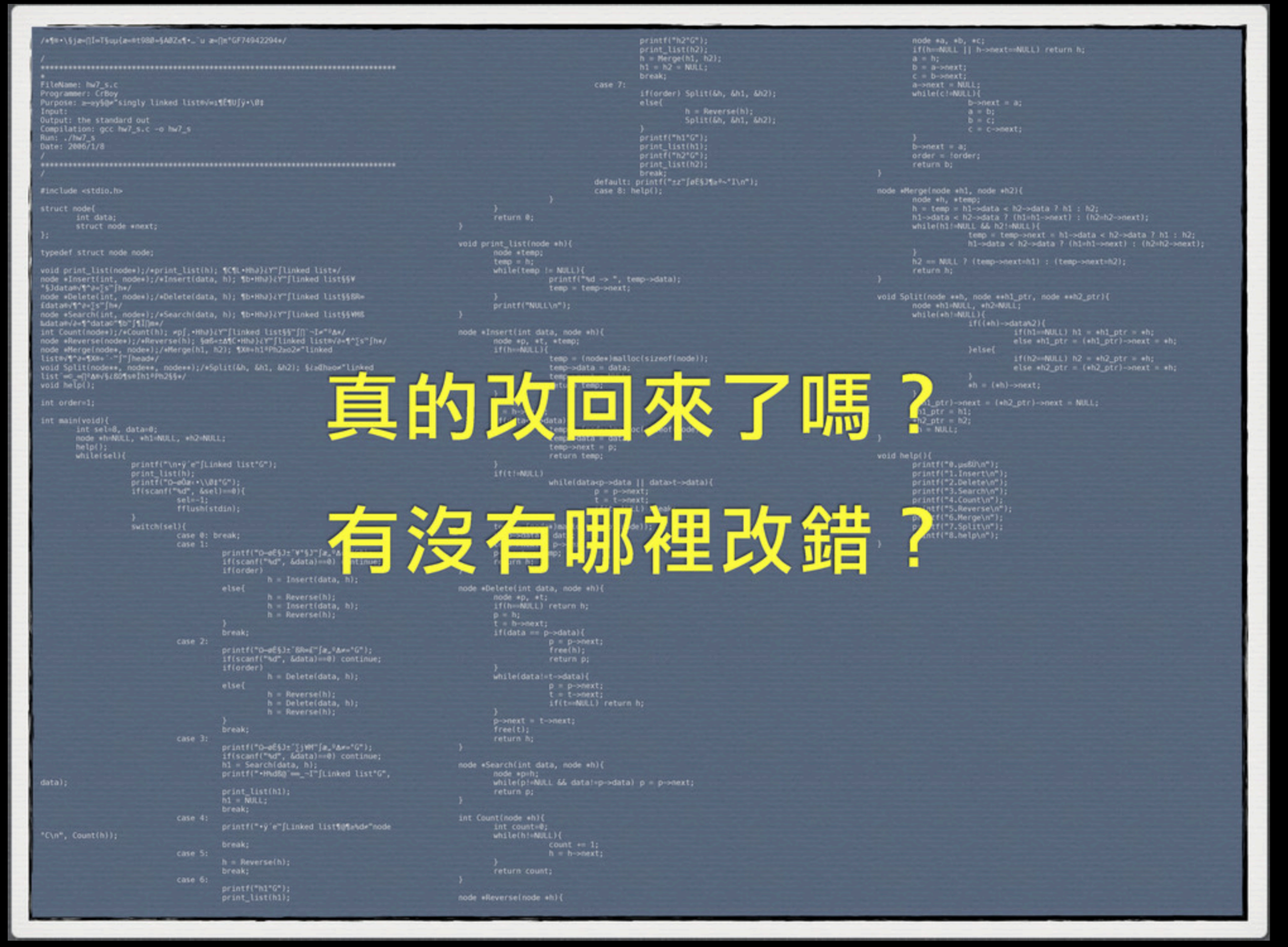
改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

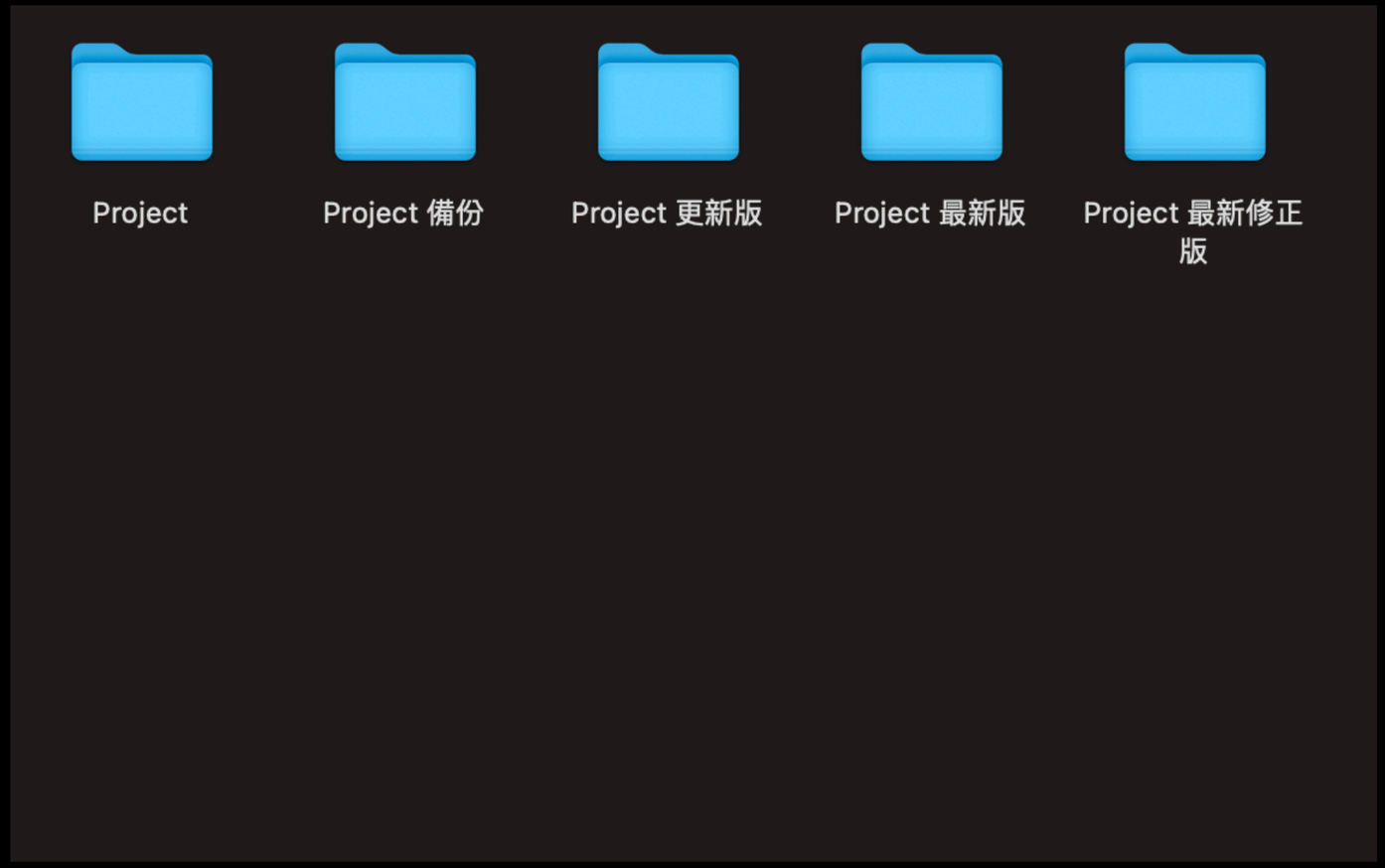
改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>



改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>



改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>



改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

然後拿 Excel 紀錄改了什麼？

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

V3	`if(battery.temperature_dotK > 3231){ //如果充電中溫度 > 50 Celsius`		變數明明小於3231，條件式卻成立	在比high byte的時候中斷發生，更新了low byte的值，導致條件成立	在while的開頭加了程式碼防止race condition	`StopSMBus();`
						`battery.voltage_mV = temp_voltage;`
						`battery.current_mA = temp_current;`
						`if(temp_temperature > temp_temperature2){`
						`  battery.temperature_dotK = temp_temperature;`
						`}`
						`else{`
						`  battery.temperature_dotK = temp_temperature2;`
						`}`
						`battery.stateOfCharge = temp_soc;`
						`StartSMBus();`
							1. 溫度會忽高忽低，此情況發生在溫度正往下掉，比如溫度現在為0C00，CPU先抓high byte 0C，溫度往下掉變為0BFE，中斷發生，抓到FE，所以最終結果是拿0CFE(3326)去比3231，導致條件成立

	`if(battery.voltage_mV < 21000){`
	`  charging_strategy = PWM_25;`
	`  IMR_min(60);//設計時60分鐘`
	`  StartTMR();  //開始設計時`
	`  led_case = PRE_CHARGE_MODE;`
	`  charger_state = PWM_CHARGING;`
	`}`
	`else{`
	`  charging_strategy = PWM_100;`
	`  charger_state = CHARGING;`
	`}`
	`CHARGER_TurnOnTheCharger();`
		`if(battery.voltage_mV < 21000){`
		`  charging_strategy = PWM_25;`
		`  IMR_min(60);//設計時60分鐘`
		`  StartTMR();  //開始設計時`
		`  led_case = PRE_CHARGE_MODE;`
		`  charger_state = PWM_CHARGING;`
		`}`
		`else{`
		`  charging_strategy = PWM_100;`
		`  charger_state = CHARGING;`
		`}`
		`CHARGER_TurnOnTheCharger();`
			多餘	在一開始已經打開充電器了，不用再打開一次	縮減程式碼	此程式碼不能修改，因為delay完5秒後，充電器的狀態不一定是開著，可能是關閉，所以還要再開一次	

	`case CHECKING_BATTERY:`
	`  if(battery.voltage_mV < 10800){`
	`  led_case = BATTERY_VOLTAGE_TOO_LOW;`
	`  charger_state = VOLTAGE_ERROR;`
	`  }`
		`if(battery.voltage_mV < 10800){`
		`  charging_strategy = PWM_0;`
		`  CHARGER_TurnOffTheCharger();`
		`  led_case = BATTERY_VOLTAGE_TOO_LOW;`
		`  charger_state = VOLTAGE_ERROR;`
		`}`
			沒有把充電器關掉	粗心阿幹	加入程式碼		

	`if(battery.temperature_dotK < 2731) led_case = TEMP_TOO_LOW_INITIAL;  //起充溫度 < 0 Celsius`		低於0度卻沒顯示燈號	溫度不能抓0x08	改抓0x4a與0x4b		1. 不要看Smart Battery spec，看9313的pdf，並且以通訊盒抓的位址為準

	`case 1:`
	`  filter.rawData_Current[k] = ADC_GetVoltage();`
	`  I2C_Master_Read(OZ9313, CURRENT, &filter.rawData_Current[k], 2);`
	`I2C_Master_Read(OZ9313, 0x03, &batMode, 2);`
	`if( batMode & (1 << 10) != 0){`
	`  I2C_Master_Read(OZ9313, CURRENT, &variable, 4);`
	`  filter.data_Current[k/2] = variable >> 8;`
	`}`
	`else{`
	`  I2C_Master_Read(OZ9313, CURRENT, &filter.rawData_Current[k], 2);`
	`}`
	`break;`
			抓出來的電流值異常	SBS裡面規定，read block時第一個回傳的是byte count	修改程式碼，以配合SBS	把BatMoe放到filter.h裡面，在filter的struct裡新增32bit的array放電流值	1. 利用ADC讀值的方式依然行不通，還是要解決

新增	`if(charger_state == PWM_CHARGING`
	`|| charger_state == CHARGING){`
	`  DELAY_ms(5000);`
		新增 `if(charger_state == PWM_CHARGING`
		`|| charger_state == CHARGING){`
		`  if(battery.current_mA <= 0){`
		`  CHARGER_TurnOffTheCharger();`
		`  charging_strategy = PWM_0;`
			如果Fuse燒斷，充電器依然顯示充電中	充電中沒有同時判斷電流值>0	在防止race condition的code上面加入第一段，	Delay要5秒，電流值才能算完，3秒不夠	目前只能用這種方式，如果用在interrupt裡面加入flag的方式，會有問題。
							因為主程式是把state切換到CHARGING後，"馬上"判斷有沒有電流。因為時間上需要等充電器啟動+OZ8920的ADC SCAN+OZ9313的換算與interrupt的通訊，所以一定要一段時間後才能執行此判斷條件。

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

一個人都搞成這樣了...

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

那如果是一個團隊呢？

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

改自：Code Smart, Don't Code hard BY 畢玉泉(CrBoy) <crboy@crboy.net>

工程師的救星 - 版本控制

在開始之前...

Git / GitHub

What’s different?

首先...

- 註冊 GitHub 帳號
- 安裝 Git，並且加到環境變數當中。

第一次使用你需要...

git config

```
git config --global user.name <your name>
git config --global user.email <your email>
git config --global core.editor <編輯器>

git config --global -e
```

其他環境設定

git config

```
git config --global color.ui true

git config --global alias.lg "log --color --graph --all --
pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset
%s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --
abbrev-commit --"
```

查看環境設定

git config

```
git config --list
git config user.name
```

global / local

git config --global / git config --local

git init

開始版本控制

追蹤檔案

git add

```
git add <files>
git add main.cpp
git add *.cpp
git add .
```

git reset <file>

取消追蹤檔案

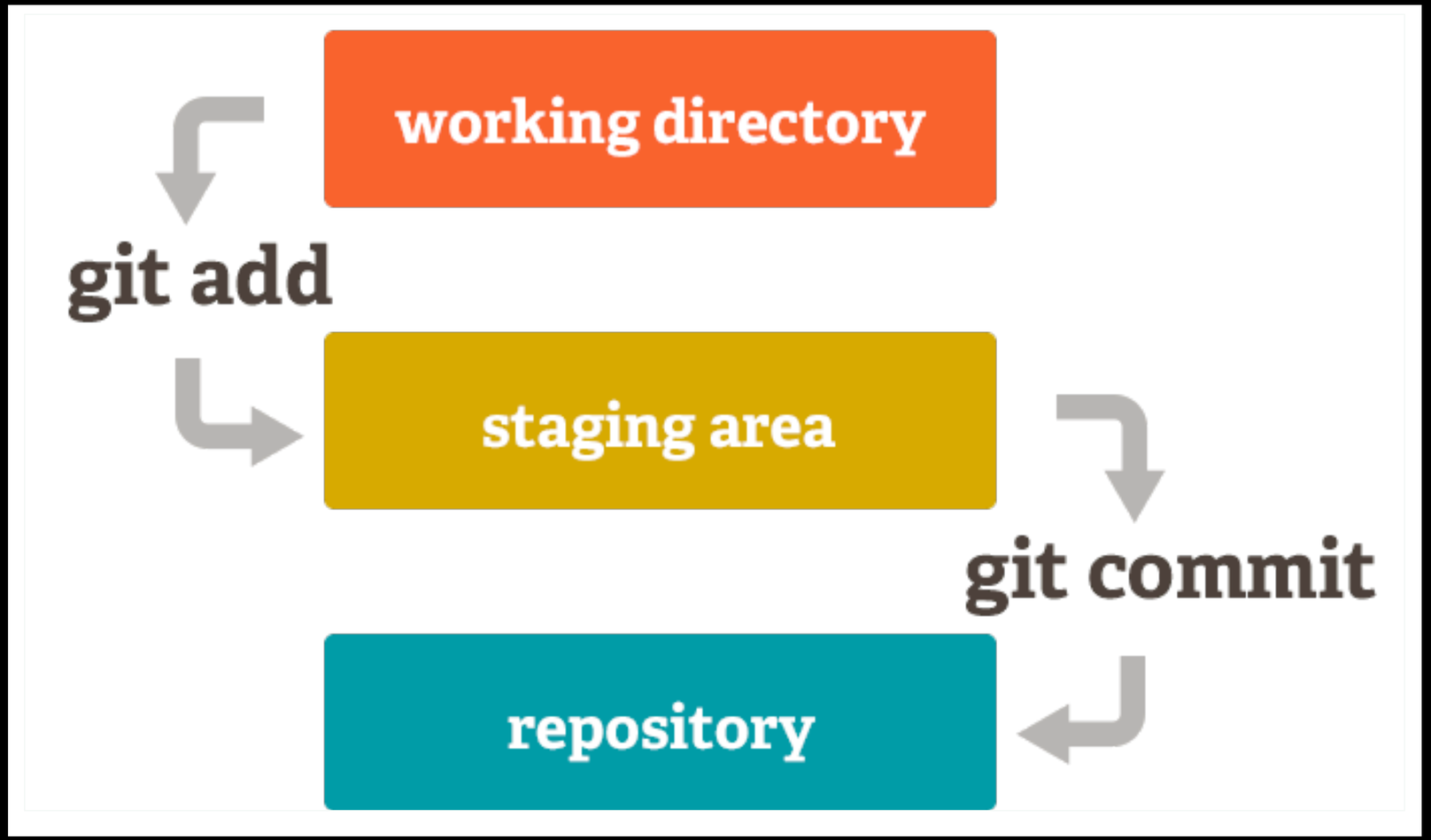
git status

查看目前狀態

提交一個新的版本

git commit

```
git commit
git commit -m <commit message>
git commit -a
```



Source: https://git-scm.com/about/staging-area

git diff

查看與上個版本的差異 / 查看與追蹤檔案的差異

git push <url> <branch>

推送 repo

git clone <url>

從遠端複製一個 git 儲存庫

git pull

拉取 repo

咪路 <mail@mirumo.org />