

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？

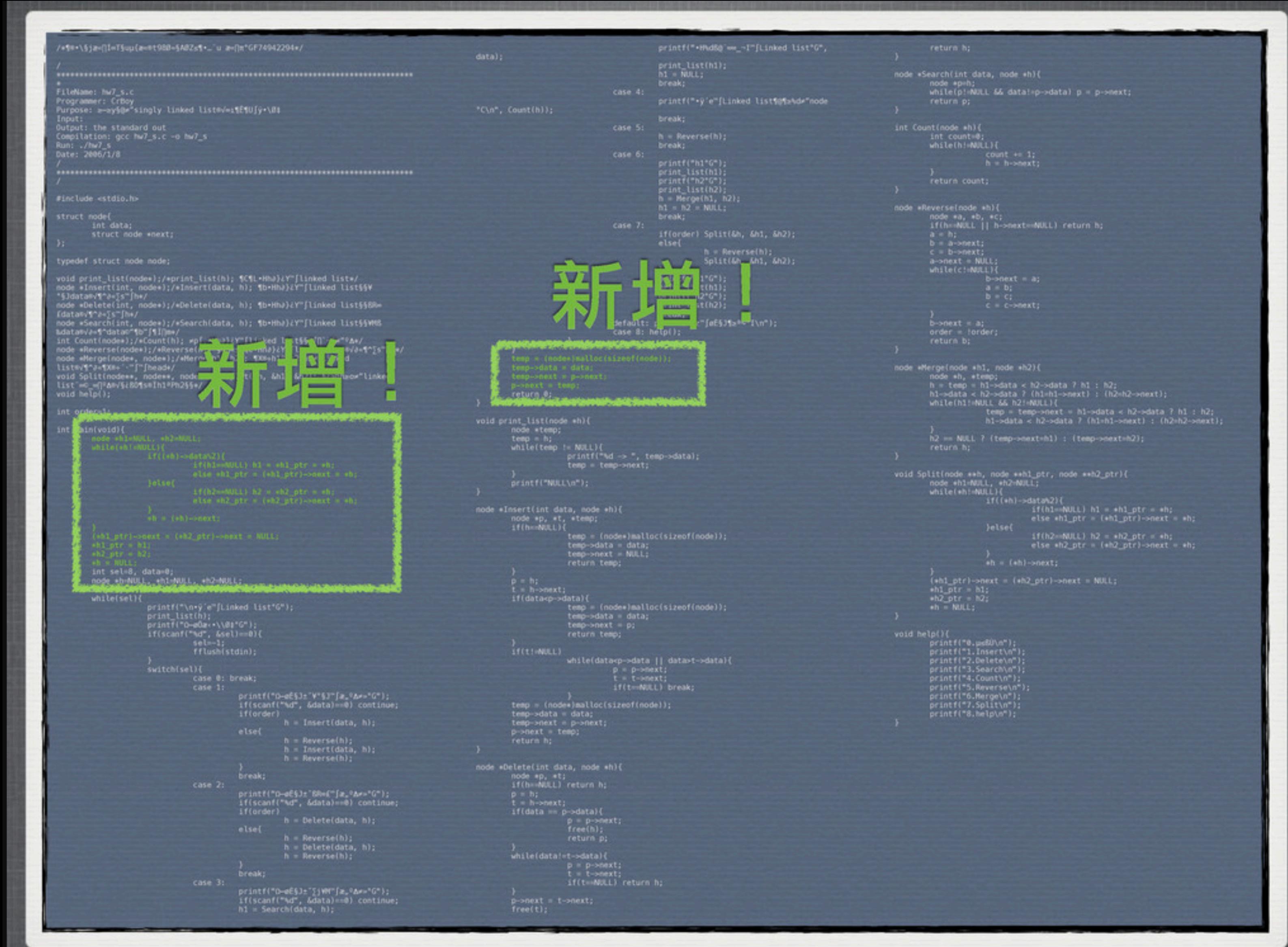
專題 / 中小型專案開發

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>



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>

```

/*$*$/5)a-[]I=TSup(a=8t988-5A8Zs$- u .m[G74942294*/.
/
*****.
* FileName: hm7_s.c
* Programmer: CrBoy
* Purpose: a=yy99 singly linked list@/w1tE4Ufy\01
* Input:
* Output: the standard out
* Compilation: gcc hm7_s.c -o hm7_s
* Run: ./hm7_s
* Date: 2006/1/8
*/
*****.

#include <stdio.h>

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

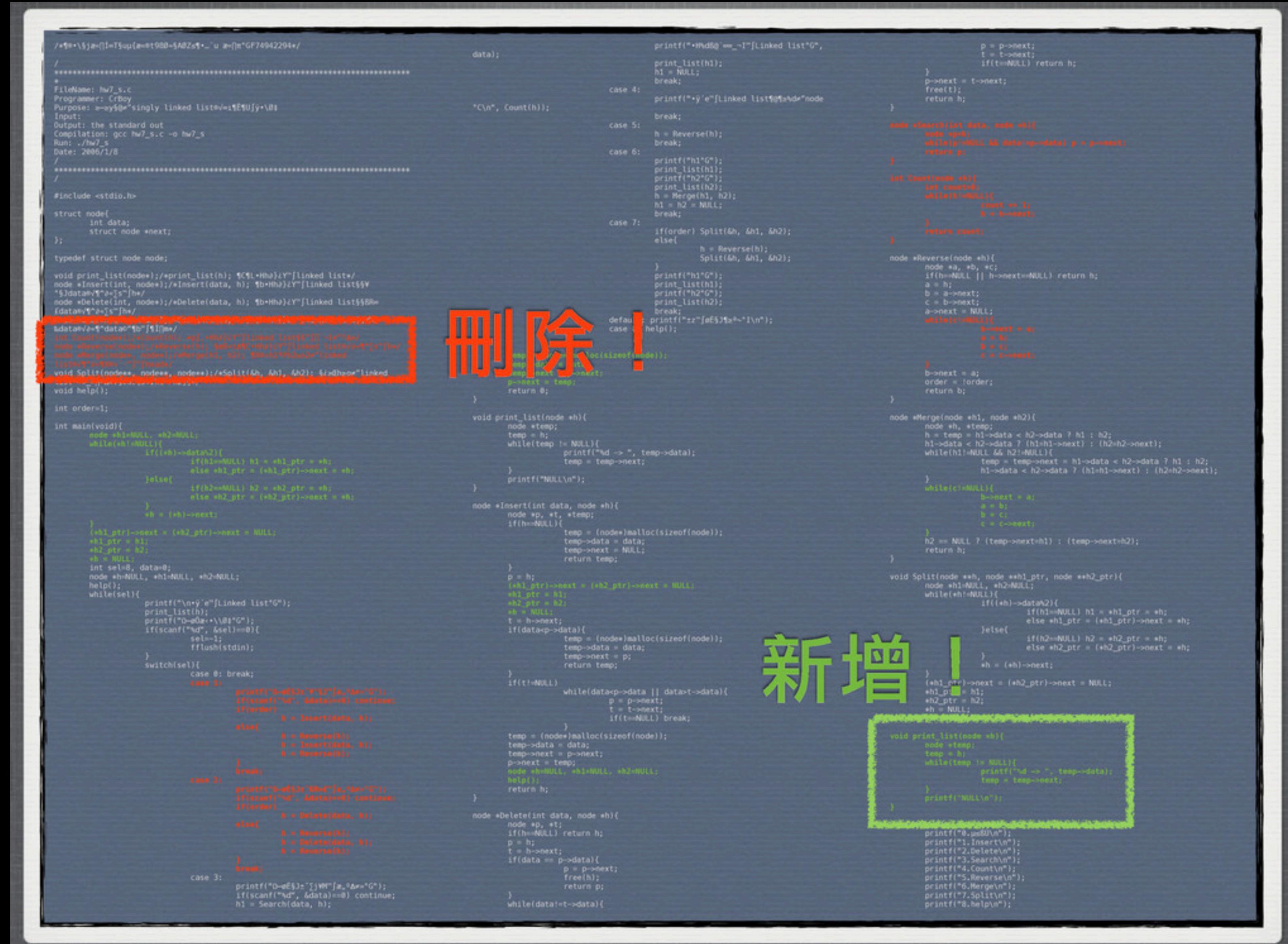
typedef struct node node;

void print_list(node*); /*print_list(h); *C$!-Hh@{Y*[linked list@/
node *Insert(int, node*); /*Insert(data, h); *b-Hh@{Y*[linked list@/
"i$data"Y*+a$*j$*h@/
node *Delete(int, node*); /*Delete(data, h); *b-Hh@{Y*[linked list@/
"i$*a$*i$*h@/
node *Search(int, node*); /*Search(data, h); *b-Hh@{Y*[linked list@/
"i$*a$*i$*h@/
int Count(node*); /*Count(h); *p{, -Hh@{Y*[linked list@/
"i$*a$*i$*h@/
node *Reverse(node*); /*Reverse(h); *b-Hh@{Y*[linked list@/
"i$*a$*i$*h@/
node *Merge(node*, node*); /*Merge(h1, h2); *b-Hh@{Y*[linked list@/
"i$*a$*i$*h@/
list@/y*+x*+i$*h@/
void Split(node*, node*, node*); /*Split(h, h1, h2); *cadHao@linked
list@=a$*b@{y*+x*+i$*h@/
void help(); /*

int order=1;

int main(void){
    node *h1=NULL, *h2=NULL;
    while(h1!=NULL){
        if(h1->data<0){
            if(h1->next==NULL) h1 = h1->ptr;
            else h1->ptr = (h1->ptr)->next = h1;
        }else{
            if(h2->next==NULL) h2 = h2->ptr;
            else h2->ptr = (h2->ptr)->next = h2;
        }
        h = (h1->next);
        (h1->ptr)->next = (h2->ptr)->next = NULL;
        h1_ptr = h1;
        h2_ptr = h2;
        h = NULL;
        int sel=8, data;
        node *h=NULL, *h1=NULL, *h2=NULL;
        help();
        while(sel){
            printf("0.Insert\n1.Delete\n2.Search\n3.Count\n4.Reverse\n5.Merge\n6.Help\n7.Quit\n");
            if(sel<0 || sel>7) sel=1;
            fflush(stdin);
        }
        switch(sel){
            case 0: break;
            case 1:
                printf("0-<0> 1->0> [a,>ae->0");
                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("0-<0> 1->0> [a,>ae->0");
                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:
                if((scanf("%d", &data)==0) continue;
                h1 = Search(data, h);
            }
        }
    }
}
*****.

*****.
* 刪除 !
* 1. Delete
* 2. Insert
* 3. Search
* 4. Reverse
* 5. Merge
* 6. Help
* 7. Quit
* 8. Count
* 9. Reverse
* 10. Merge
* 11. Insert
* 12. Delete
* 13. Search
* 14. Count
* 15. Reverse
* 16. Merge
* 17. Help
* 18. Count
* 19. Reverse
* 20. Merge
* 21. Insert
* 22. Delete
* 23. Search
* 24. Count
* 25. Reverse
* 26. Merge
* 27. Help
* 28. Count
* 29. Reverse
* 30. Merge
* 31. Insert
* 32. Delete
* 33. Search
* 34. Count
* 35. Reverse
* 36. Merge
* 37. Help
* 38. Count
* 39. Reverse
* 40. Merge
* 41. Insert
* 42. Delete
* 43. Search
* 44. Count
* 45. Reverse
* 46. Merge
* 47. Help
* 48. Count
* 49. Reverse
* 50. Merge
* 51. Insert
* 52. Delete
* 53. Search
* 54. Count
* 55. Reverse
* 56. Merge
* 57. Help
* 58. Count
* 59. Reverse
* 60. Merge
* 61. Insert
* 62. Delete
* 63. Search
* 64. Count
* 65. Reverse
* 66. Merge
* 67. Help
* 68. Count
* 69. Reverse
* 70. Merge
* 71. Insert
* 72. Delete
* 73. Search
* 74. Count
* 75. Reverse
* 76. Merge
* 77. Help
* 78. Count
* 79. Reverse
* 80. Merge
* 81. Insert
* 82. Delete
* 83. Search
* 84. Count
* 85. Reverse
* 86. Merge
* 87. Help
* 88. Count
* 89. Reverse
* 90. Merge
* 91. Insert
* 92. Delete
* 93. Search
* 94. Count
* 95. Reverse
* 96. Merge
* 97. Help
* 98. Count
* 99. Reverse
* 100. Merge
* 101. Insert
* 102. Delete
* 103. Search
* 104. Count
* 105. Reverse
* 106. Merge
* 107. Help
* 108. Count
* 109. Reverse
* 110. Merge
* 111. Insert
* 112. Delete
* 113. Search
* 114. Count
* 115. Reverse
* 116. Merge
* 117. Help
* 118. Count
* 119. Reverse
* 120. Merge
* 121. Insert
* 122. Delete
* 123. Search
* 124. Count
* 125. Reverse
* 126. Merge
* 127. Help
* 128. Count
* 129. Reverse
* 130. Merge
* 131. Insert
* 132. Delete
* 133. Search
* 134. Count
* 135. Reverse
* 136. Merge
* 137. Help
* 138. Count
* 139. Reverse
* 140. Merge
* 141. Insert
* 142. Delete
* 143. Search
* 144. Count
* 145. Reverse
* 146. Merge
* 147. Help
* 148. Count
* 149. Reverse
* 150. Merge
* 151. Insert
* 152. Delete
* 153. Search
* 154. Count
* 155. Reverse
* 156. Merge
* 157. Help
* 158. Count
* 159. Reverse
* 160. Merge
* 161. Insert
* 162. Delete
* 163. Search
* 164. Count
* 165. Reverse
* 166. Merge
* 167. Help
* 168. Count
* 169. Reverse
* 170. Merge
* 171. Insert
* 172. Delete
* 173. Search
* 174. Count
* 175. Reverse
* 176. Merge
* 177. Help
* 178. Count
* 179. Reverse
* 180. Merge
* 181. Insert
* 182. Delete
* 183. Search
* 184. Count
* 185. Reverse
* 186. Merge
* 187. Help
* 188. Count
* 189. Reverse
* 190. Merge
* 191. Insert
* 192. Delete
* 193. Search
* 194. Count
* 195. Reverse
* 196. Merge
* 197. Help
* 198. Count
* 199. Reverse
* 200. Merge
* 201. Insert
* 202. Delete
* 203. Search
* 204. Count
* 205. Reverse
* 206. Merge
* 207. Help
* 208. Count
* 209. Reverse
* 210. Merge
* 211. Insert
* 212. Delete
* 213. Search
* 214. Count
* 215. Reverse
* 216. Merge
* 217. Help
* 218. Count
* 219. Reverse
* 220. Merge
* 221. Insert
* 222. Delete
* 223. Search
* 224. Count
* 225. Reverse
* 226. Merge
* 227. Help
* 228. Count
* 229. Reverse
* 230. Merge
* 231. Insert
* 232. Delete
* 233. Search
* 234. Count
* 235. Reverse
* 236. Merge
* 237. Help
* 238. Count
* 239. Reverse
* 240. Merge
* 241. Insert
* 242. Delete
* 243. Search
* 244. Count
* 245. Reverse
* 246. Merge
* 247. Help
* 248. Count
* 249. Reverse
* 250. Merge
* 251. Insert
* 252. Delete
* 253. Search
* 254. Count
* 255. Reverse
* 256. Merge
* 257. Help
* 258. Count
* 259. Reverse
* 260. Merge
* 261. Insert
* 262. Delete
* 263. Search
* 264. Count
* 265. Reverse
* 266. Merge
* 267. Help
* 268. Count
* 269. Reverse
* 270. Merge
* 271. Insert
* 272. Delete
* 273. Search
* 274. Count
* 275. Reverse
* 276. Merge
* 277. Help
* 278. Count
* 279. Reverse
* 280. Merge
* 281. Insert
* 282. Delete
* 283. Search
* 284. Count
* 285. Reverse
* 286. Merge
* 287. Help
* 288. Count
* 289. Reverse
* 290. Merge
* 291. Insert
* 292. Delete
* 293. Search
* 294. Count
* 295. Reverse
* 296. Merge
* 297. Help
* 298. Count
* 299. Reverse
* 300. Merge
* 301. Insert
* 302. Delete
* 303. Search
* 304. Count
* 305. Reverse
* 306. Merge
* 307. Help
* 308. Count
* 309. Reverse
* 310. Merge
* 311. Insert
* 312. Delete
* 313. Search
* 314. Count
* 315. Reverse
* 316. Merge
* 317. Help
* 318. Count
* 319. Reverse
* 320. Merge
* 321. Insert
* 322. Delete
* 323. Search
* 324. Count
* 325. Reverse
* 326. Merge
* 327. Help
* 328. Count
* 329. Reverse
* 330. Merge
* 331. Insert
* 332. Delete
* 333. Search
* 334. Count
* 335. Reverse
* 336. Merge
* 337. Help
* 338. Count
* 339. Reverse
* 340. Merge
* 341. Insert
* 342. Delete
* 343. Search
* 344. Count
* 345. Reverse
* 346. Merge
* 347. Help
* 348. Count
* 349. Reverse
* 350. Merge
* 351. Insert
* 352. Delete
* 353. Search
* 354. Count
* 355. Reverse
* 356. Merge
* 357. Help
* 358. Count
* 359. Reverse
* 360. Merge
* 361. Insert
* 362. Delete
* 363. Search
* 364. Count
* 365. Reverse
* 366. Merge
* 367. Help
* 368. Count
* 369. Reverse
* 370. Merge
* 371. Insert
* 372. Delete
* 373. Search
* 374. Count
* 375. Reverse
* 376. Merge
* 377. Help
* 378. Count
* 379. Reverse
* 380. Merge
* 381. Insert
* 382. Delete
* 383. Search
* 384. Count
* 385. Reverse
* 386. Merge
* 387. Help
* 388. Count
* 389. Reverse
* 390. Merge
* 391. Insert
* 392. Delete
* 393. Search
* 394. Count
* 395. Reverse
* 396. Merge
* 397. Help
* 398. Count
* 399. Reverse
* 400. Merge
* 401. Insert
* 402. Delete
* 403. Search
* 404. Count
* 405. Reverse
* 406. Merge
* 407. Help
* 408. Count
* 409. Reverse
* 410. Merge
* 411. Insert
* 412. Delete
* 413. Search
* 414. Count
* 415. Reverse
* 416. Merge
* 417. Help
* 418. Count
* 419. Reverse
* 420. Merge
* 421. Insert
* 422. Delete
* 423. Search
* 424. Count
* 425. Reverse
* 426. Merge
* 427. Help
* 428. Count
* 429. Reverse
* 430. Merge
* 431. Insert
* 432. Delete
* 433. Search
* 434. Count
* 435. Reverse
* 436. Merge
* 437. Help
* 438. Count
* 439. Reverse
* 440. Merge
* 441. Insert
* 442. Delete
* 443. Search
* 444. Count
* 445. Reverse
* 446. Merge
* 447. Help
* 448. Count
* 449. Reverse
* 450. Merge
* 451. Insert
* 452. Delete
* 453. Search
* 454. Count
* 455. Reverse
* 456. Merge
* 457. Help
* 458. Count
* 459. Reverse
* 460. Merge
* 461. Insert
* 462. Delete
* 463. Search
* 464. Count
* 465. Reverse
* 466. Merge
* 467. Help
* 468. Count
* 469. Reverse
* 470. Merge
* 471. Insert
* 472. Delete
* 473. Search
* 474. Count
* 475. Reverse
* 476. Merge
* 477. Help
* 478. Count
* 479. Reverse
* 480. Merge
* 481. Insert
* 482. Delete
* 483. Search
* 484. Count
* 485. Reverse
* 486. Merge
* 487. Help
* 488. Count
* 489. Reverse
* 490. Merge
* 491. Insert
* 492. Delete
* 493. Search
* 494. Count
* 495. Reverse
* 496. Merge
* 497. Help
* 498. Count
* 499. Reverse
* 500. Merge
* 501. Insert
* 502. Delete
* 503. Search
* 504. Count
* 505. Reverse
* 506. Merge
* 507. Help
* 508. Count
* 509. Reverse
* 510. Merge
* 511. Insert
* 512. Delete
* 513. Search
* 514. Count
* 515. Reverse
* 516. Merge
* 517. Help
* 518. Count
* 519. Reverse
* 520. Merge
* 521. Insert
* 522. Delete
* 523. Search
* 524. Count
* 525. Reverse
* 526. Merge
* 527. Help
* 528. Count
* 529. Reverse
* 530. Merge
* 531. Insert
* 532. Delete
* 533. Search
* 534. Count
* 535. Reverse
* 536. Merge
* 537. Help
* 538. Count
* 539. Reverse
* 540. Merge
* 541. Insert
* 542. Delete
* 543. Search
* 544. Count
* 545. Reverse
* 546. Merge
* 547. Help
* 548. Count
* 549. Reverse
* 550. Merge
* 551. Insert
* 552. Delete
* 553. Search
* 554. Count
* 555. Reverse
* 556. Merge
* 557. Help
* 558. Count
* 559. Reverse
* 560. Merge
* 561. Insert
* 562. Delete
* 563. Search
* 564. Count
* 565. Reverse
* 566. Merge
* 567. Help
* 568. Count
* 569. Reverse
* 570. Merge
* 571. Insert
* 572. Delete
* 573. Search
* 574. Count
* 575. Reverse
* 576. Merge
* 577. Help
* 578. Count
* 579. Reverse
* 580. Merge
* 581. Insert
* 582. Delete
* 583. Search
* 584. Count
* 585. Reverse
* 586. Merge
* 587. Help
* 588. Count
* 589. Reverse
* 590. Merge
* 591. Insert
* 592. Delete
* 593. Search
* 594. Count
* 595. Reverse
* 596. Merge
* 597. Help
* 598. Count
* 599. Reverse
* 600. Merge
* 601. Insert
* 602. Delete
* 603. Search
* 604. Count
* 605. Reverse
* 606. Merge
* 607. Help
* 608. Count
* 609. Reverse
* 610. Merge
* 611. Insert
* 612. Delete
* 613. Search
* 614. Count
* 615. Reverse
* 616. Merge
* 617. Help
* 618. Count
* 619. Reverse
* 620. Merge
* 621. Insert
* 622. Delete
* 623. Search
* 624. Count
* 625. Reverse
* 626. Merge
* 627. Help
* 628. Count
* 629. Reverse
* 630. Merge
* 631. Insert
* 632. Delete
* 633. Search
* 634. Count
* 635. Reverse
* 636. Merge
* 637. Help
* 638. Count
* 639. Reverse
* 640. Merge
* 641. Insert
* 642. Delete
* 643. Search
* 644. Count
* 645. Reverse
* 646. Merge
* 647. Help
* 648. Count
* 649. Reverse
* 650. Merge
* 651. Insert
* 652. Delete
* 653. Search
* 654. Count
* 655. Reverse
* 656. Merge
* 657. Help
* 658. Count
* 659. Reverse
* 660. Merge
* 661. Insert
* 662. Delete
* 663. Search
* 664. Count
* 665. Reverse
* 666. Merge
* 667. Help
* 668. Count
* 669. Reverse
* 670. Merge
* 671. Insert
* 672. Delete
* 673. Search
* 674. Count
* 675. Reverse
* 676. Merge
* 677. Help
* 678. Count
* 679. Reverse
* 680. Merge
* 681. Insert
* 682. Delete
* 683. Search
* 684. Count
* 685. Reverse
* 686. Merge
* 687. Help
* 688. Count
* 689. Reverse
* 690. Merge
* 691. Insert
* 692. Delete
* 693. Search
* 694. Count
* 695. Reverse
* 696. Merge
* 697. Help
* 698. Count
* 699. Reverse
* 700. Merge
* 701. Insert
* 702. Delete
* 703. Search
* 704. Count
* 705. Reverse
* 706. Merge
* 707. Help
* 708. Count
* 709. Reverse
* 710. Merge
* 711. Insert
* 712. Delete
* 713. Search
* 714. Count
* 715. Reverse
* 716. Merge
* 717. Help
* 718. Count
* 719. Reverse
* 720. Merge
* 721. Insert
* 722. Delete
* 723. Search
* 724. Count
* 725. Reverse
* 726. Merge
* 727. Help
* 728. Count
* 729. Reverse
* 730. Merge
* 731. Insert
* 732. Delete
* 733. Search
* 734. Count
* 735. Reverse
* 736. Merge
* 737. Help
* 738. Count
* 739. Reverse
* 740. Merge
* 741. Insert
* 742. Delete
* 743. Search
* 744. Count
* 745. Reverse
* 746. Merge
* 747. Help
* 748. Count
* 749. Reverse
* 750. Merge
* 751. Insert
* 752. Delete
* 753. Search
* 754. Count
* 755. Reverse
* 756. Merge
* 757. Help
* 758. Count
* 759. Reverse
* 760. Merge
* 761. Insert
* 762. Delete
* 763. Search
* 764. Count
* 765. Reverse
* 766. Merge
* 767. Help
* 768. Count
* 769. Reverse
* 770. Merge
* 771. Insert
* 772. Delete
* 773. Search
* 774. Count
* 775. Reverse
* 776. Merge
* 777. Help
* 778. Count
* 779. Reverse
* 780. Merge
* 781. Insert
* 782. Delete
* 783. Search
* 784. Count
* 785. Reverse
* 786. Merge
* 787. Help
* 788. Count
* 789. Reverse
* 790. Merge
* 791. Insert
* 792. Delete
* 793. Search
* 794. Count
* 795. Reverse
* 796. Merge
* 797. Help
* 798. Count
* 799. Reverse
* 800. Merge
* 801. Insert
* 802. Delete
* 803. Search
* 804. Count
* 805. Reverse
* 806. Merge
* 807. Help
* 808. Count
* 809. Reverse
* 810. Merge
* 811. Insert
* 812. Delete
* 813. Search
* 814. Count
* 815. Reverse
* 816. Merge
* 817. Help
* 818. Count
* 819. Reverse
* 820. Merge
* 821. Insert
* 822. Delete
* 823. Search
* 824. Count
* 825. Reverse
* 826. Merge
* 827. Help
* 828. Count
* 829. Reverse
* 830. Merge
* 831. Insert
* 832. Delete
* 833. Search
* 834. Count
* 835. Reverse
* 836. Merge
* 837. Help
* 838. Count
* 839. Reverse
* 840. Merge
* 841. Insert
* 842. Delete
* 843. Search
* 844. Count
* 845. Reverse
* 846. Merge
* 847. Help
* 848. Count
* 849. Reverse
* 850. Merge
* 851. Insert
* 852. Delete
* 853. Search
* 854. Count
* 855. Reverse
* 856. Merge
* 8
```



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>



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>

```

/*
* $Id: hw7_s.c 1 2006-01-08 14:22:04 CrBoy Exp $
* singly linked list
* CrBoy
* Purpose: singly linked list
* 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 *h);
node *Insert(int, node *h);
node *Delete(int, node *h);
node *Search(int, node *h);
int Count(node *h);
node *Reverse(node *h);
node *Merge(node *h1, node *h2);
node *Split(node *h, node **h1, node **h2);
void help();
int main(void){
    node *h1=NULL, *h2=NULL;
    while(h1!=NULL || h2!=NULL){
        if((h1->data)>(h2->data)){
            if(h1->data==0) h1 = h1->next;
            else h1_ptr = (h1->data)->next;
        }else{
            if(h2->data==0) h2 = h2->next;
            else h2_ptr = (h2->data)->next;
        }
        h = (h->next);
    }
    (h1_ptr)->next = (h2_ptr)->next = NULL;
    h1_ptr = h1;
    h2_ptr = h2;
    h = NULL;
    int sel=0, data=0;
    node *h=NULL, *h1=NULL, *h2=NULL;
    help();
    while(sel<10){
        printf("\n0-插入\n1-删除\n2-查找\n3-输出\n4-计数\n5-反转\n6-合并\n7-拆分\n8-帮助\n");
        switch(sel){
            case 0: break;
            case 1:
                printf("0-插入\n1-删除\n2-查找\n3-输出\n4-计数\n5-反转\n6-合并\n7-拆分\n8-帮助\n");
                if(scanf("%d", &data)==0) continue;
                if(sel==0) h = Insert(data, h);
                else h = Reverse(h);
                h = Insert(data, h);
                h = Reverse(h);
            break;
            case 2:
                printf("0-插入\n1-删除\n2-查找\n3-输出\n4-计数\n5-反转\n6-合并\n7-拆分\n8-帮助\n");
                if(scanf("%d", &data)==0) continue;
                if(sel==0) h = Delete(data, h);
                else h = Reverse(h);
                h = Delete(data, h);
                h = Reverse(h);
            break;
            case 3:
                printf("0-插入\n1-删除\n2-查找\n3-输出\n4-计数\n5-反转\n6-合并\n7-拆分\n8-帮助\n");
                if(scanf("%d", &data)==0) continue;
                h1 = Search(data, h);
        }
    }
}
int Count(node *h){
    int count=0;
    while(h!=NULL){
        count += 1;
        h = h->next;
    }
    return count;
}
node *Search(int data, node *h){
    node *p=h;
    while(p!=NULL && data!=p->data) p = p->next;
    return p;
}
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 = b;
    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;
    }
    b->next = a;
    order = b;
    return b;
}
node *Split(node *h, node **h1, node **h2){
    node *h1=NULL, *h2=NULL;
    while(h!=NULL){
        if((h->data)>2){
            if(h1==NULL) h1 = h->next;
            else h1_ptr = (h1->data)->next;
        }else{
            if(h2==NULL) h2 = h->next;
            else h2_ptr = (h2->data)->next;
        }
        h = (h->next);
    }
    (h1_ptr)->next = (h2_ptr)->next = NULL;
    h1_ptr = h1;
    h2_ptr = h2;
    h = NULL;
}
void help(){
    printf("0-插入\n1-删除\n2-查找\n3-输出\n4-计数\n5-反转\n6-合并\n7-拆分\n8-帮助\n");
}
void print_list(node *h){
    node *temp;
    temp = h;
    while(temp!=NULL){
        printf("%d\n", temp->data);
        temp = temp->next;
    }
}
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;
}
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;
        if(t==NULL) break;
        temp = (node*)malloc(sizeof(node));
        temp->data = data;
        temp->next = p;
        return temp;
    }
    if(t!=NULL)
        while(data!=t->data){
            p = p->next;
            t = t->next;
            if(t==NULL) break;
        }
    if(data==t->data){
        p = p->next;
        t = t->next;
        if(t==NULL) return h;
        p->next = t->next;
        free(t);
        return p;
    }
    while(data!=t->data){
        p = p->next;
        t = t->next;
        if(t==NULL) return h;
        p->next = t->next;
        free(t);
    }
    p->next = t->next;
    free(t);
    return h;
}

```

改回來改回來改回來！



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

```

/*$*•\$|a-[]|I=I$up(a=8t988$ABZa\$-~ u =|[n`GF74942294*/
```

```

+
FileName: hw7_s.c
Programmer: CrBoy
Purpose: a->y$* singly linked list*/s$E6U[y->0s
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$L+Hba}Y"linked list*/
node *Insert(int, node*); /*Insert(data, h); *B+Hba}Y"linked list$*/
"3$data\*"\*a\*is\*h\*
node *Delete(int, node*); /*Delete(data, h); *B+Hba}Y"linked list$B$R=
"data\*"\*a\*is\*h\*
node *Search(int, node*); /*Search(data, h); *B+Hba}Y"linked list$B$R
adata\*"\*data\*"\*f\*1]p\*
int Count(node*); /*Count(h); *pf+Hba}Y"linked list$J\*"\*a\*is\*h\*
node *Reverse(node*); /*Reverse(h); *B+Hba}Y"linked list\*a\*is\*h\*
node *Merge(node*, node*); /*Merge(h1, h2); *X\*h1\*Ph2ao2*linked
list\*"\*a\*X\*s\*"\*]*/head\*
void Split(node*, node*); /*Split(h, h1, h2); *S\*adbaow\*linked
list =-\*a\*W\*s\*B\*s\*h1\*h2$*/s
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>e\*Linked list\*G");
        print_list();
        printf("0-e\*-\*0t\*G");
        if(scanf("0", &data)==0){
            sel=-1;
            fflush(stdin);
        }
        switch(sel){
            case 0: break;
            case 1:
                printf("0-e\$)\+V\$"\*a\*is\*h\*;
                if(scanf("0d", &data)==0) continue;
                if(order)
                    h = Insert(data, h);
                else
                    h = Reverse(h);
                h = Insert(data, h);
                h = Reverse(h);
            break;
            case 2:
                printf("0-e\$)\+B\*d\*"\*a\*is\*G\*";
                if(scanf("0d", &data)==0) continue;
                if(order)
                    h = Delete(data, h);
                else
                    h = Reverse(h);
                h = Delete(data, h);
                h = Reverse(h);
            break;
            case 3:
                printf("0-e\$)\+I\*M\*"\*a\*is\*G\*";
                if(scanf("0d", &data)==0) continue;
                h1 = Search(data, h);
                printf("0-Mud\$0\*-\*l\*linked list\*G\*",
data);
                print_list(h1);
                h1 = NULL;
            break;
            case 4:
                printf("0-y\*e\*linked list@M\*nde\*node
break;
            case 5:
                h = Reverse(h);
            break;
            case 6:
                printf("h1\*G\*");
                print_list(h1);
        }
    }
    "C\n", Count(h));
}
```

```

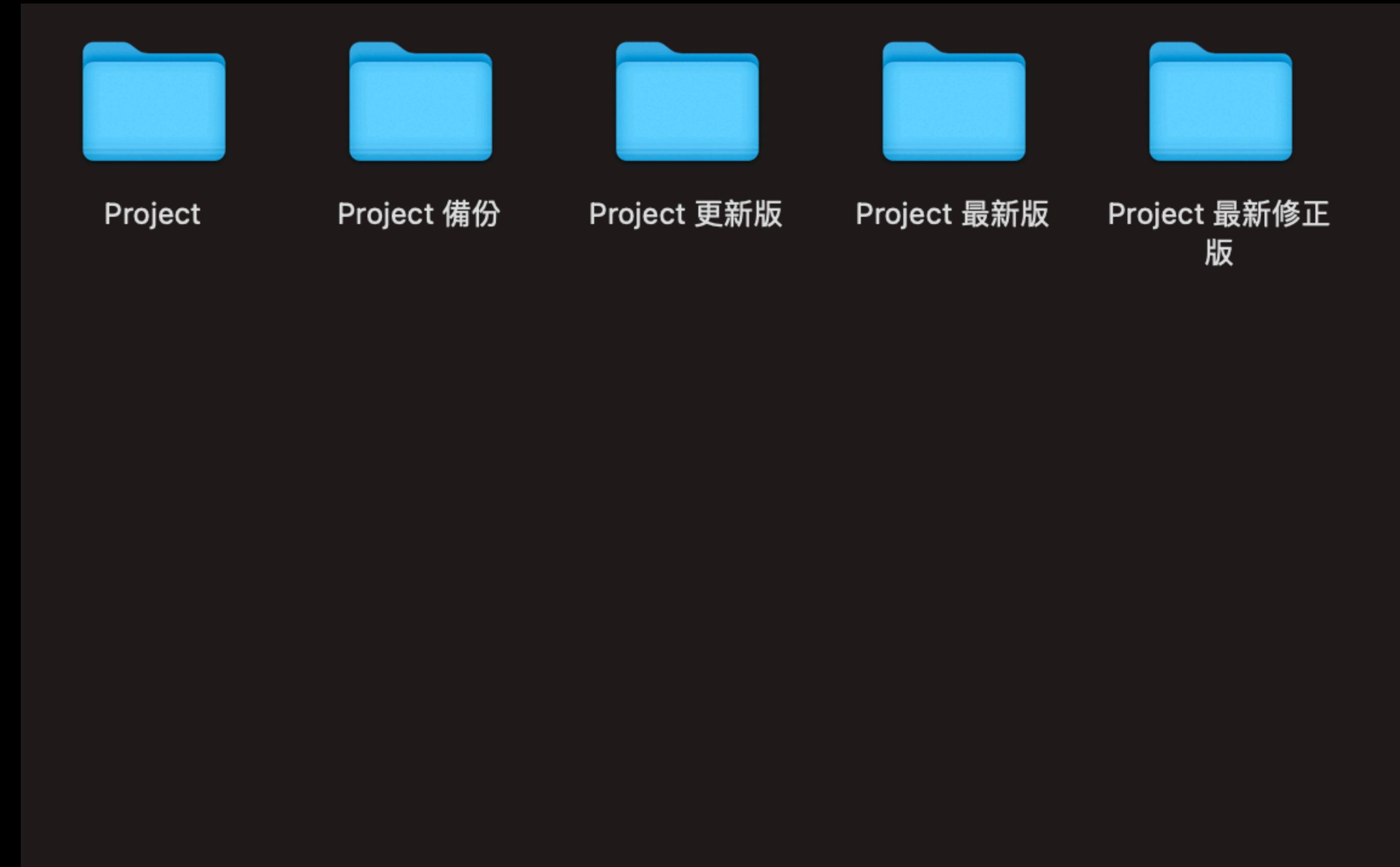
    case 7:
        printf("h2\*G\*");
        print_list(h2);
        h = Merge(h1, h2);
        h1 = h2 = NULL;
        break;
    if(order) Split(h, h1, h2);
    else
        h = Reverse(h);
        Split(h, h1, h2);
    printf("h1\*G\*");
    print_list(h1);
    printf("h2\*G\*");
    print_list(h2);
    break;
default: printf("zz\*je\$%\*a\*1\*n");
    }
}
case 8: help();
}
}

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)->data2{
            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.us0\*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");
}
```

真的改回來了嗎？
有沒有哪裡改錯？



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	<pre>if(battery.temperature_dotK > 3231) //如果充電中溫度 > 50 Celsius</pre>		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	<pre>StopIMBus(); 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; StartIMBus();</pre>	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
	<pre>if(battery.voltage_mV < 21000){ charging_strategy = PWM_25; TMR_min(60); //7.5毫秒的時間 StartIMR(); led_case = PRE_CHARGE_MODE; charger_state = PWM_CHARGING; } else{ charging_strategy = PWM_100; charger_state = CHARGING; } CHARGER_TurnOnTheCharger();</pre>	<pre>if(battery.voltage_mV < 21000){ charging_strategy = PWM_25; TMR_min(60); //7.5毫秒的時間 StartIMR(); led_case = PRE_CHARGE_MODE; charger_state = PWM_CHARGING; } else{ charging_strategy = PWM_100; charger_state = CHARGING; } CHARGER_TurnOnTheCharger();</pre>		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
	<pre>case CHECKING_BATTERY: if(battery.voltage_mV < 10800){ led_case = BATTERY_VOLTAGE_TOO_LOW; charger_state = VOLTAGE_ERROR; } }</pre>	<pre>if(battery.voltage_mV < 10800){ charging_strategy = PWM_0; CHARGER_TurnOffTheCharger(); led_case = BATTERY_VOLTAGE_TOO_LOW; charger_state = VOLTAGE_ERROR; }</pre>		沒有把充電器關掉	粗心阿幹	加入程式碼	
	<pre>if(battery.temperature_dotK < 2731) led_case = TEMP_TOO_LOW_INITIAL; //起充溫度 < 0 Celsius</pre>		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
	<pre>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;</pre>		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	<pre>if(charger_state == PWM_CHARGING charger_state == CHARGING){ DELAY_ms(5000); }</pre>	新增	<pre>if(charger_state == PWM_CHARGING charger_state == CHARGING){ if(battery.current_mA < 0){ CHARGER_TurnOffTheCharger(); charging_strategy = PWM_0; led_case = PWM_ERROR; } }</pre>	如果Fuse燒斷，充電器依然顯示在電流中	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判斷條件。

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>



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

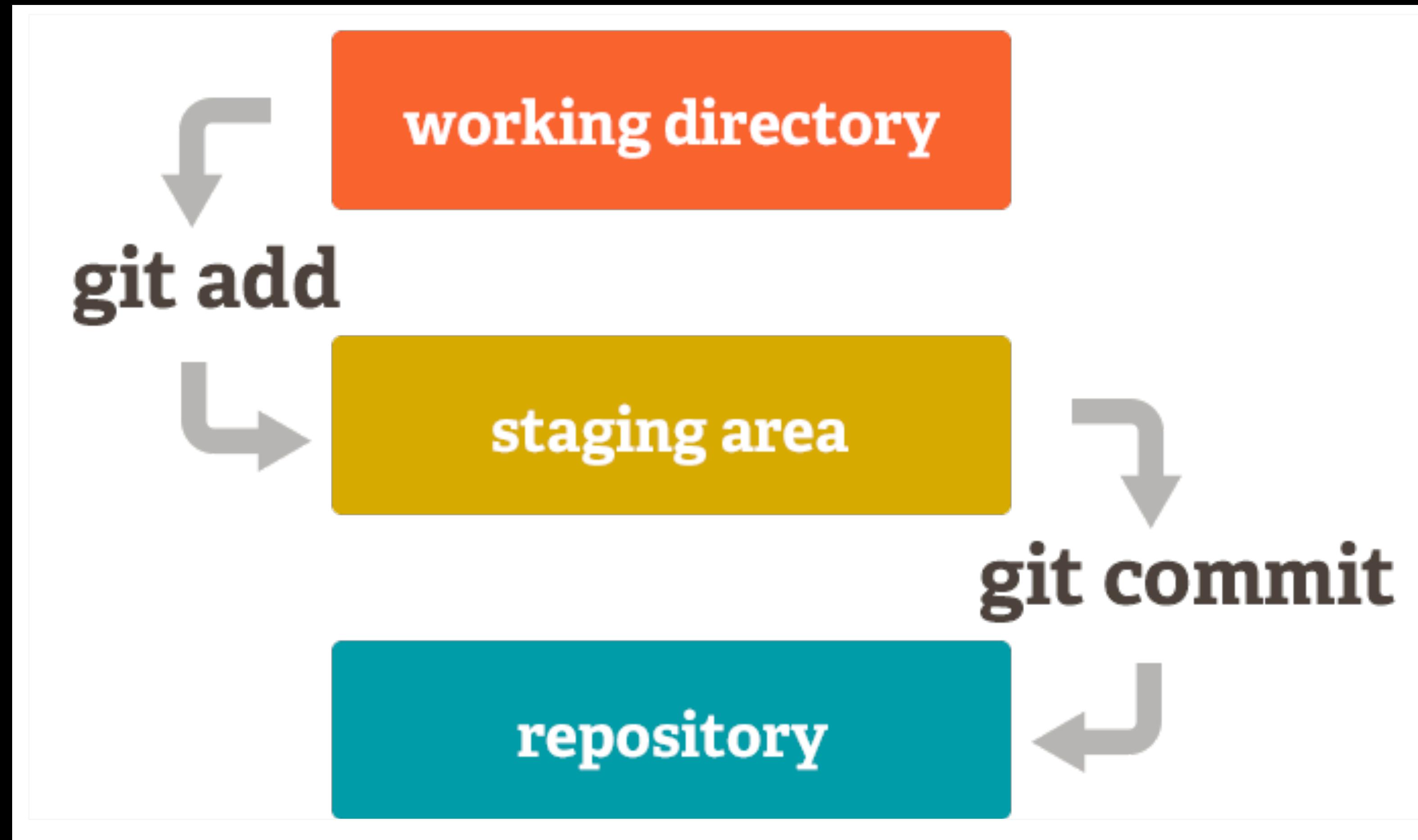
提交一個新的版本

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 />