

ex1:

1.sudo find ./ -name "bin" -exec ls -l {} \;

```
kivygogh — find • sudo — 80x24
Last login: Thu Nov 23 12:52:22 on ttys000
goudandeMacBook-Pro:~ goudan$ sudo find / -name "bin" -exec ls -l {} \;
>Password:
total 134240
-rwxr-xr-x  4 root  wheel      925  7 16 08:20 2to3-
lrwxr-xr-x  1 root  wheel      74 11 14 01:26 2to3-2.7 -> ../../System/Libr
ary/Frameworks/Python.framework/Versions/2.7/bin/2to3-2.7
-rwxr-xr-x  1 root  wheel    55056 10 26 00:37 AssetCacheLocatorUtil
-rwxr-xr-x  1 root  wheel    53472 10 26 00:37 AssetCacheManagerUtil
-rwxr-xr-x  1 root  wheel    48288 10 26 00:37 AssetCacheTetheratorUtil
-rwxr-xr-x  1 root  wheel    18320 10 26 00:37 BuildStrings
-rwxr-xr-x  1 root  wheel    18288 10 26 00:37 CpMac
-rwxr-xr-x  1 root  wheel    18288 10 26 00:37 DeRez
-rwxr-xr-x  1 root  wheel    18320 10 26 00:37 GetFileInfo
-rwxr-xr-x  1 root  wheel    74032 10 26 00:37 IOAccelMemory
-rwxr-xr-x  1 root  wheel    18304 10 26 00:37 MergePef
-rwxr-xr-x  1 root  wheel    18288 10 26 00:37 MvMac
-rwxr-xr-x  1 root  wheel    18304 10 26 00:37 ResMerger
-rwxr-xr-x  1 root  wheel    18288 10 26 00:37 Rez
-rwxr-xr-x  1 root  wheel    18288 10 26 00:37 RezDet
-rwxr-xr-x  1 root  wheel    18304 10 26 00:37 RezWack
-rwxr-xr-x  1 root  wheel    18304 10 26 00:37 SetFile
-rwxr-xr-x  1 root  wheel    18304 10 26 00:37 SplitForks
-rwxr-xr-x  1 root  wheel    18304 10 26 00:37 UnRezWack
```

2.find ./ -type f -size +3k

```
kivygogh — -bash — 80x24
goudandeMacBook-Pro:~ goudan$ find . -type f -size +3k
./svg/index.html
./svg/.DS_Store
./svg/node_modules/babylon/package.json
./svg/node_modules/babylon/lib/util/identifier.js
./svg/node_modules/babylon/lib/plugins/flow.js
./svg/node_modules/babylon/lib/plugins/jsx/xhtml.js
./svg/node_modules/babylon/lib/plugins/jsx/index.js
./svg/node_modules/babylon/lib/tokenizer/types.js
./svg/node_modules/babylon/lib/tokenizer/index.js
./svg/node_modules/babylon/lib/parser/lval.js
./svg/node_modules/babylon/lib/parser/statement.js
./svg/node_modules/babylon/lib/parser/comments.js
./svg/node_modules/babylon/lib/parser/expression.js
./svg/node_modules/jstsc/bin/jstsc
./svg/node_modules/jstsc/jstsc.js
./svg/node_modules/jstsc/README.md
./svg/node_modules/globals/globals.json
./svg/node_modules/postcss-cli/node_modules/chokidar/CHANGELOG.md
./svg/node_modules/postcss-cli/node_modules/chokidar/node_modules/anymatch/node_
modules/micromatch/node_modules/xtglob/index.js
./svg/node_modules/postcss-cli/node_modules/chokidar/node_modules/anymatch/node_
modules/micromatch/node_modules/xtglob/README.md
./svg/node_modules/postcss-cli/node_modules/chokidar/node_modules/anymatch/node_
```

3.find. / -exec o 000 chmod 755 {} \;

4.find ./ -name "*.c" root | xargs wc -l | sort -n >> sorted_statistics.txt

5.find ./ root | wc -l

ex2:

1. Use `pthread_create` to save threads' id, bind their thread function and send args to threads.

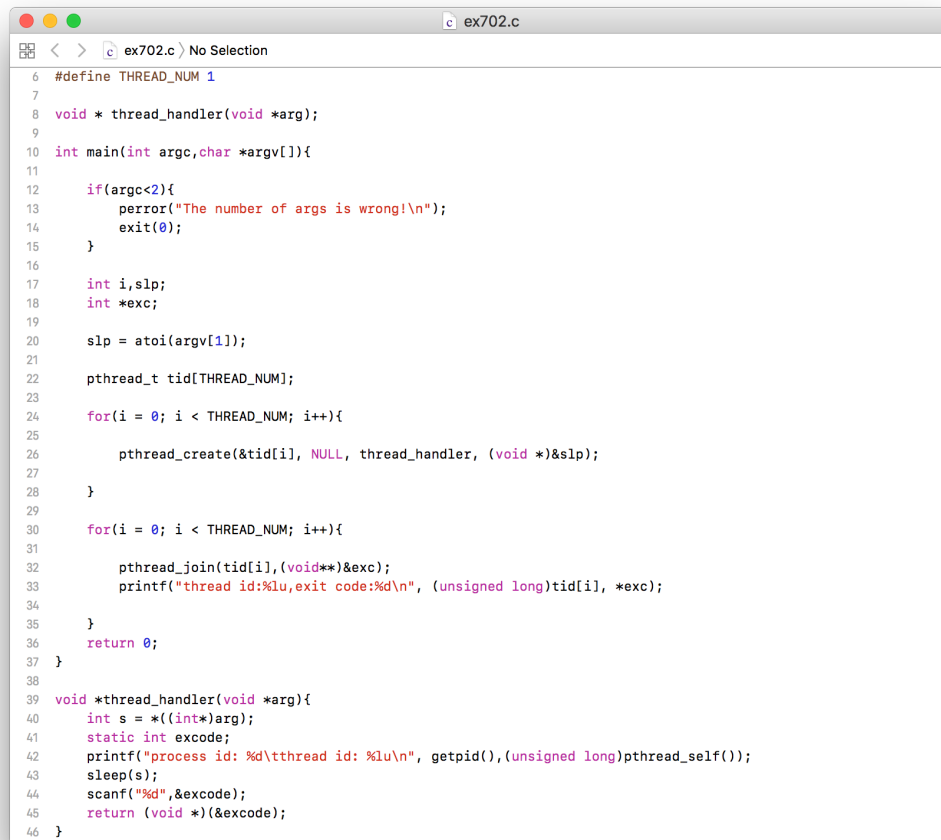
```
pthread_create(&tid[i], NULL, thread_handler, (void *)&slp);
```

2. Use `pthread_join` to wait for threads' termination, and receive their return value.

```
pthread_join(tid[i], (void **)&exc);
```

3. Use `getpid` to get pid, use `pthread_self` to get thread identifier.

```
printf("process id: %d\tthread id: %lu\n", getpid(), (unsigned long)pthread_self());
```



```
6 #define THREAD_NUM 1
7
8 void * thread_handler(void *arg);
9
10 int main(int argc, char *argv[]){
11
12     if(argc<2){
13         perror("The number of args is wrong!\n");
14         exit(0);
15     }
16
17     int i, slp;
18     int *exc;
19
20     slp = atoi(argv[1]);
21
22     pthread_t tid[THREAD_NUM];
23
24     for(i = 0; i < THREAD_NUM; i++){
25
26         pthread_create(&tid[i], NULL, thread_handler, (void *)&slp);
27     }
28
29     for(i = 0; i < THREAD_NUM; i++){
30
31         pthread_join(tid[i], (void **)&exc);
32         printf("thread id: %lu, exit code: %d\n", (unsigned long)tid[i], *exc);
33     }
34     return 0;
35 }
36
37 void *thread_handler(void *arg){
38     int s = *((int *)arg);
39     static int excode;
40     printf("process id: %d\tthread id: %lu\n", getpid(), (unsigned long)pthread_self());
41     sleep(s);
42     scanf("%d", &excode);
43     return (void *)&excode;
44 }
```

ex3:

1. Use a binary tree to store every thread's deep and thread id. Malloc sharing memory to binary tree in main thread.

Tree root;

creat_tree(root, n);

root->tid = pthread_self();

2. Create threads and share data.

root->left->deep = n-1;

pthread_create(&root->left->tid, NULL, thread_handler, root->left);

root->right->deep = n-1;

pthread_create(&root->right->tid, NULL, thread_handler, root->right);

3. Generate child threads and store data in the binary tree in handler function.

```
void *thread_handler(void *args){
    Tree t = (Tree)args;
    if(t->deep == 0)
    { t->left = NULL;
      t->right = NULL;
      exit(0);
    }else{
        pthread_create(&t->left->tid, NULL, thread_handler, t->left);
        pthread_create(&t->right->tid, NULL, thread_handler, t->right);
    }
    return NULL;
}
```