1.

struct：

```
typedef struct threadData {
    pthread_t threadId;
    FILE *fp;
    int sum;
} threadData ;
```

Initialize semaphore:

```
sem_init (&sem, 0, 1);
```

creat threads and join:

```
for (i=0; i<t; i++) {
         td[i].fp = fp;
         pthread_create (&(td[i].threadId),NULL,readFile,(void *) &td[i]);
}

    for (i=0; i<t; i++) {
         pthread_join (td[i].threadId, (void**)&retval);
         printf("id:%ul,number:%d", (unsigned int)td[i].threadId,td[i].sum);
    }
```

Thread handler:

```
static void *readFile (void *arg){
    threadData *td=(threadData *)arg;
    char *filename,*cmd;
    int retVal;
    int number=0;

    filename = (char*)malloc(sizeof(char)*L);
    cmd = (char*)malloc(sizeof(char)*L);
    while (1) {
        sem_wait (&sem);
        retVal = fscanf (td->fp, "%s", filename);
        sem_post (&sem);
        if (retVal == EOF)
            break;
        sprintf(cmd,"wc -l %s",filename);
        system(cmd);
        number++;
        sleep (1);
    }
    td->sum=number;
    return NULL;
}
```

2.

Determine if it is a file/director:

```
if [ -f $subfile ]; then
        if [ -r $subfile ]; then
```

Get size:

```
size=$(wc -c $subfile | awk '{print $1}')
```

Determine the size:

```
if [[ $size -gt 1024 ]]; then
```

3.

Get old id and compute the id

```
olduid=$(cat /etc/passwd | tail -n 1 $filename | cut -d ':' -f 4)
let uid=olduid+1
```

append line:

```
append=$user":x:"$gid":"$uid":""$name"",,,:/home/"$user":/bin/bash"
```