(1) In proc.h, added inside struct proc:

```
int page_faults;
```

(2) In proc.c, added inside allocproc():

```
p->page_faults = 0;
```

(3) In syscall.h:

```
#define SYS_getpagefaults  22
```

(4) In sysproc.c added:

```
int
sys_getpagefaults(void)
{
  struct proc *p = myproc();
  return p->page_faults;
}
```

(5) In syscall.c added:

```
extern int sys_getpagefaults(void);

[SYS_getpagefaults] sys_getpagefaults,
```

(6) In user.h added:

```
int getpagefaults(void);
```

(7) In usys.S added:

```
SYSCALL(getpagefaults)
```

(8) In vm.c added:

```
int
vmfault(pde_t *pgdir, uint va, int write)
{
  struct proc *p = myproc();
  char *mem;

  if (va >= p->sz)
    return -1;
  va = PGROUNDDOWN(va);
  if (walkpgdir(pgdir, (void *)va, 0))
    return 0;

  mem = kalloc();
  if (mem == 0)
    return -1;

  memset(mem, 0, PGSIZE);
  if (mappages(pgdir, (void *)va, PGSIZE, V2P(mem), PTE_W|PTE_U) < 0) {
    kfree(mem);
    return -1;
  }
  return 0;
}
```

(9) In trap.c added:

```
  case T_PGFLT: {
    struct proc *p = myproc();
    if(p){
      p->page_faults++;
      if(vmfault(p->pgdir, rcr2(), tf->err & 2) < 0){
        p->killed = 1;    // kill process on failure
      }
      break;
    }
  }
```

(10) Added in defs.h:

```
int         vmfault(pde_t *pgdir, uint va, int write);
```

(11) Created tlbrun.c and added the following in it:

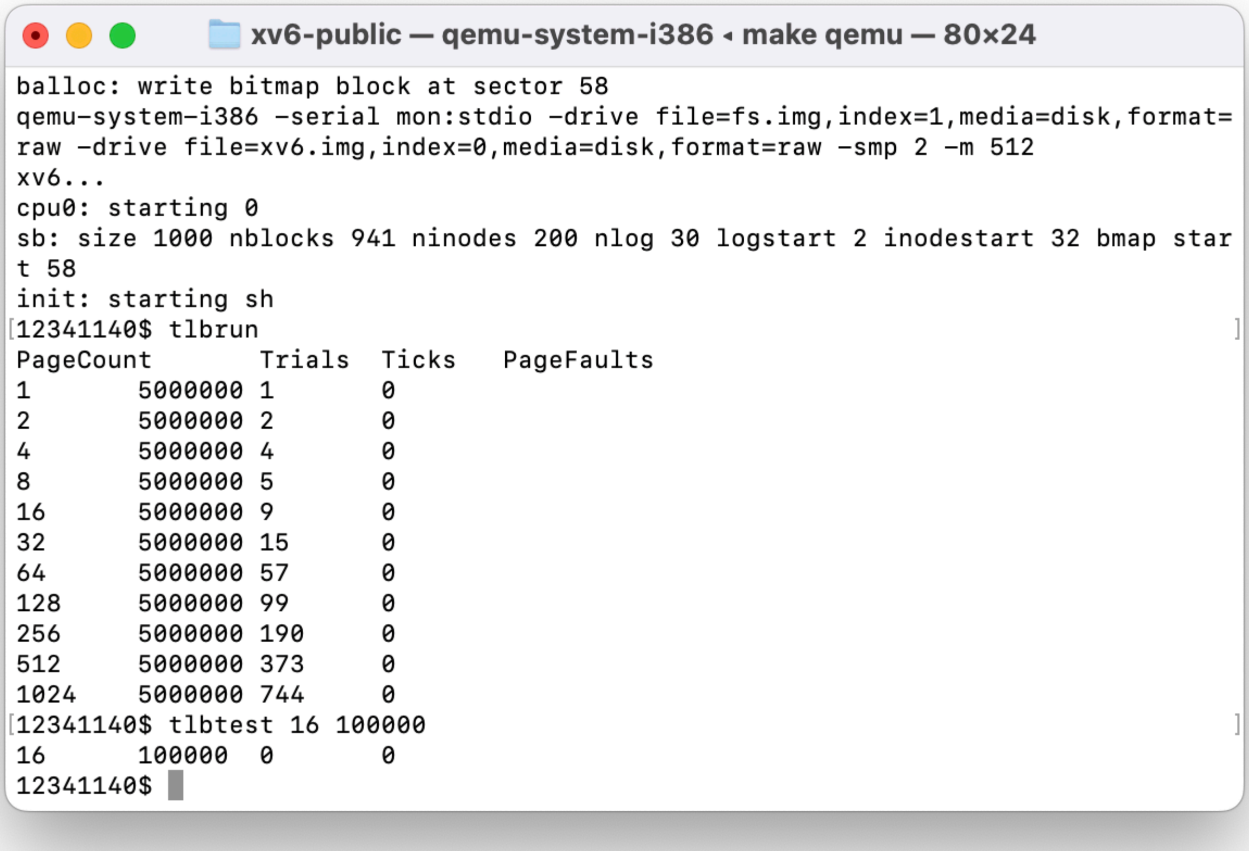
```
#include "types.h"
#include "user.h"

#define PAGESIZE 4096
#define MAXPAGES 1024

int
main(void)
{
  int jump = PAGESIZE / sizeof(int);
  printf(1, "PageCount\tTrials\tTicks\tPageFaults\n");

  for (int numpages = 1; numpages <= MAXPAGES; numpages *= 2) {
    int trials = 5000000;
    int faults_before = getpagefaults();
    int start = uptime();

    int *arr = (int*) sbrk(numpages * PAGESIZE);
    if (arr == (void*) -1)
      exit();

    for (int t = 0; t < trials; t++) {
      for (int i = 0; i < (numpages/2)*jump; i += jump) {
        arr[i] += 1;
      }
    }

    int end = uptime();
    int faults_after = getpagefaults();
    printf(1, "%d\t%d\t%d\t%d\n",
           numpages, trials, end-start, faults_after - faults_before);
  }

  exit();
}
```

(12) Created tlbtest.c and added the following in it:

```
#include "types.h"
#include "user.h"

#define PAGESIZE 4096

int
main(int argc, char *argv[])
{
  if (argc < 3) {
    printf(1, "Usage: tlbtest <pagecount> <trials>\n");
    exit();
  }

  int numpages = atoi(argv[1]);
  int trials = atoi(argv[2]);
  int jump = PAGESIZE / sizeof(int);

  int faults_before = getpagefaults();
  int start = uptime();

  int *arr = (int*) sbrk(numpages * PAGESIZE);
  if (arr == (void*) -1)
    exit();

  for (int t = 0; t < trials; t++) {
    for (int i = 0; i < (numpages/2)*jump; i += jump) {
      arr[i] += 1;
    }
  }

  int end = uptime();
  int faults_after = getpagefaults();
  printf(1, "%d\t%d\t%d\t%d\n",
        numpages, trials, end-start, faults_after - faults_before);

  exit();
}
```

(13) In Makefile, under UPROGS added:

```
    _tlbrun\
    _tlbtest\
```



```
balloc: write bitmap block at sector 58
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=
raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
12341140$ tlbrun
PageCount       Trials  Ticks   PageFaults
1       5000000 1       0
2       5000000 2       0
4       5000000 4       0
8       5000000 5       0
16      5000000 9       0
32      5000000 15      0
64      5000000 57      0
128     5000000 99      0
256     5000000 190     0
512     5000000 373     0
1024    5000000 744     0
12341140$ tlbtest 16 100000
16      100000  0       0
12341140$ 
```

Screenshot of the Terminal