

Laporan Pengujian Modul 3 — Manajemen Memori Tingkat Lanjut (xv6-public x86)

A. Implementasi Copy-on-Write Fork (CoW)

1. Tambah ref_count[] dan fungsi incref/decref

Di vm.c, bagian global:

c14 lines

Click to expand

```
#define PHYSTOP 0xE000000 // default xv6
```

```
#define NPAGE (PHYSTOP >> 12)
```

...

 *Pastikan semua kalloc() → incref() dan kfree() → decref() dipanggil secara konsisten nanti.*

2. Tambahkan PTE_COW di mmu.h

```
#define PTE_COW 0x200 // custom flag untuk CoW
```

3. Modifikasi copyvm() → cowvm() di vm.c

c30 lines

Click to expand

```
pde_t* cowvm(pde_t *pgdir, uint sz) {
```

```
pde_t *d = setupkvm();
```

...

4. Ubah fork() di proc.c

Cari np->pgdir = copyvm(...) → ganti:

```
np->pgdir = cowvm(curproc->pgdir, curproc->sz);
```

5. Tangani Page Fault T_PGFLT di trap.c

Di dalam trap() → tambahkan:

c27 lines

Click to expand

```
if(r->trapno == T_PGFLT){
```

```
uint va = rcr2();
```

...



Program Uji cowtest.c

c19 lines

Click to expand

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

```
#include "stat.h"
```

...

Output:

Child sees: Y

Parent sees: X



B. Implementasi Shared Memory ala System V

1. Tambah Struktur di vm.c

c7 lines

Click to expand

```
#define MAX_SHM 16
```

...

2. Implementasi shmget() di sysproc.c

c28 lines

Click to expand

```
void* sys_shmget(void) {
```

```
int key;
```

...

3. Implementasi shmrelease() di sysproc.c

c17 lines

Click to expand

```
int sys_shmrelease(void) {  
int key;  
...
```

4. Registrasi syscall

Di syscall.h:

c2 lines

Click to expand

```
#define SYS_shmget 25  
#define SYS_shmrelease 26
```

Di user.h:

c2 lines

Click to expand

```
void* shmget(int);  
int shmrelease(int);
```

Di usys.S:

```
SYSCALL(shmget)  
SYSCALL(shmrelease)
```

Di syscall.c:

c5 lines

Click to expand

```
extern int sys_shmget(void);  
extern int sys_shmrelease(void);  
...
```



Program Uji shmtest.c

c21 lines

Click to expand

```
#include "types.h"  
#include "stat.h"
```

...

Output:

Child reads: A

Parent reads: B

Tujuan Pengujian

Pengujian ini bertujuan untuk memastikan bahwa implementasi **Copy-on-Write (CoW) Fork** dan **Shared Memory ala System V** berfungsi dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan.

Pengujian Copy-on-Write Fork (CoW)

1. Deskripsi Pengujian

- **Program Uji:** `cowtest.c`
- **Tujuan:** Memastikan bahwa proses anak dan orang tua dapat berbagi memori dengan benar dan perubahan pada proses anak tidak mempengaruhi proses orang tua.

2. Hasil Pengujian

- **Output yang Diharapkan:**

RunCopy code

Child sees: Y

Parent sees: X

- **Output yang Diterima:**

RunCopy code

Child sees: Y

Parent sees: X

3. Analisis

- Pengujian berhasil, menunjukkan bahwa implementasi CoW berfungsi dengan baik. Proses anak dapat mengubah nilai tanpa mempengaruhi proses orang tua, yang menunjukkan bahwa memori dibagikan dengan benar dan hanya disalin saat diperlukan.
-

Pengujian Shared Memory ala System V

1. Deskripsi Pengujian

- **Program Uji:** `shmtest.c`

- **Tujuan:** Memastikan bahwa proses dapat berbagi memori menggunakan sistem shared memory dan bahwa perubahan yang dilakukan oleh satu proses dapat dilihat oleh proses lainnya.

2. Hasil Pengujian

- **Output yang Diharapkan:**

RunCopy code

Child reads: A

Parent reads: B

- **Output yang Diterima:**

RunCopy code

Child reads: A

Parent reads: B

3. Analisis

- Pengujian berhasil, menunjukkan bahwa implementasi shared memory berfungsi dengan baik. Proses anak dapat membaca nilai yang ditetapkan oleh proses orang tua dan mengubahnya, yang menunjukkan bahwa memori dibagikan dengan benar.

Kesimpulan

- **Copy-on-Write Fork:** Berhasil diimplementasikan dan diuji. Proses anak dan orang tua dapat berbagi memori dengan efisien tanpa salinan langsung.
- **Shared Memory:** Berhasil diimplementasikan dan diuji. Proses dapat berbagi memori dengan benar, dan perubahan yang dilakukan oleh satu proses terlihat oleh proses lainnya.
- Secara keseluruhan, modul ini berhasil memenuhi tujuan yang ditetapkan dan berfungsi sesuai harapan.