

README

tictactoe_ipc (src)

Clone this repository using `git clone` or download a snapshot [here](#)

Run `make` to build.

Run `./ttt --server` and `./ttt --client` on different terminal session to try it out.

Penjelasan Program

Program ini (`tictactoe_ipc`) adalah sebuah permainan tictactoe yang dapat dijalankan oleh dua pemain. Kedua pemain berjalan pada dua buah proses yang berbeda (Player 1 sebagai server, Player 2 sebagai client). Kedua proses saling berkomunikasi melalui IPC dengan metode Shared Memory. Metode Shared Memory disini menggunakan API POSIX `mmap()` ([wiki](#)). `mmap()` digunakan untuk *memory-map* suatu file atau device (lebih lanjut: `man 2 mmap`), dalam hal ini `mmap()` dapat juga digunakan sebagai fasilitas untuk melakukan IPC dengan shared memory.

Penggunaan `mmap()` disini mirip seperti menggunakan `malloc()`. Mengalokasikan memori untuk pointer ke sebuah struct `ttt` sebesar `sizeof(ttt)`, atau mengambil alamat memori jika alamat tersebut telah teralokasi ([src](#)), dengan proteksi `ReadWrite`, dan flags `MAP_SHARED`, agar objek dan alamat memori tersebut dapat di share dan digunakan oleh proses lain.

Program cukup menggunakan tiga buah abstraksi fungsi pada `tttshm.h` untuk mendapatkan alamat/object dari shared memory, yaitu `open_and_init_shm()` (untuk mengalokasi memori dan mengambil alamat object), `open_shm()` (mengambil alamat object saja), dan `close_shm()` (untuk meng-unmap object `t` di memory). Return object dari ketiga fungsi ini berupa object `t` yang dimap ke dalam memory.

Penjelasan Source

- `tttshm.{h,c}`
 - `tttshm.h` ([src](#)): Mendefinisikan beberapa function header, konstanta dan flags yang akan digunakan pada `tttshm.c`
 - `tttshm.c` ([src](#)): Implementasi abstraksi fungsi untuk mengalokasi dan mendapatkan alamat object `t` pada shared memory.

```
* static ttt *_open_shm(int flags)
```

Jika flags `SHM_INIT`, maka fungsi tersebut akan membuat file `SHMFILE` baru untuk di-map ke memory, mengalokasi alamat shared memory untuk object `t` sebesar `sizeof(ttt)`, dan menginisialisasi nilai awal untuk object `t`. Fungsi akan me-return alamat object `t`.

Jika flags `SHM_OPEN`, fungsi hanya akan membaca file `SHMFILE` untuk mengidentifikasi shared mapped-memory object, lalu mendapatkan alamat shared memory dari object `t`. Fungsi akan me-return alamat `t`.

Jika flags `SHM_END`, fungsi hanya akan membaca file `SHMFILE` untuk mengidentifikasi shared mapped-memory object, lalu meng-unmap object pada memory. Fungsi akan me-return `NULL`.

* `ttt *open_and_init_shm()`: alias dari `_open_shm(SHM_INIT)`

* `ttt *open_shm()`: alias dari `_open_shm(SHM_OPEN)`

* `ttt *close_shm()`: alias dari `_open_shm(SHM_INIT)`

- `ttt.{h,c}`

- `ttt.h` (src): Mendefinisikan beberapa function header, `struct ttt`, dan konstanta yang akan digunakan pada `ttt.c`

- `ttt.c` (src): Implementasi dari `tictactoe_ipc`

- * `static void print_board(ttt *t)`: Print board tictactoe dari object `t`

- * `static int check_board(ttt *t, int player)`: Mengecek apakah board sudah ada pemenang atau draw

- * `static void cpu_move(ttt *t, int player)`: AI untuk CPU Player

- * `static int get_input_and_check(ttt *t, int player, player_t p_type)`: Mendapatkan input langkah dari player (jika `Player == Human`, maka membaca `stdin`, jika `Player == PC`, menggunakan `cpu_move()`), dan mengecek input langkah tersebut.

- * `static void player_loop(ttt *t, int player, player_t p_type)`: Player loop. This is the game! Memasukkan input untuk meletakkan “pion”-nya, menunggu giliran pemain lain, pengumuman pemenang, dan draw, juga mengakhiri game jika sudah selesai.

- * `void server_start(player_t p_type)`: Player 1, mengalokasikan shared memory object `t`, menginisilaisasi game state, dan menunggu client (Player 2). Kalau Player 2 sudah terhubung, masuk ke state `player_loop`.

- * `void client_start(player_t p_type)`: Player 2, mendapatkan alamat shared memory object `t`, mengecek apakah player masih berjumlah 1 (Belum ada client yang terhubung). Jika Player 2 terhubung ke Player 1 (server), masuk ke state `player_loop`.

- `main.c` ([src](#))

Main Program dari tictactoe, membaca argumen program saat program dijalankan, dari argumen yang diberikan, program mengerti akan berjalan sebagai server/client ataupun player merupakan manusia atau cpu/pc.

Argumen yang dapat diberikan antara lain:

Untuk argumen 1:

`--server`: Menjalankan program sebagai server.

`--client`: Menjalankan program sebagai client.

Untuk argumen 2:

`--pc`: Sebagai pemain CPU (bukan manusia)

Saat proses menunggu, digunakan `usleep()` selama 30ms, agar tidak terlalu “busy-wait” yang menghabiskan proses CPU.

Cara Menjalankan Program

Seperti pada bagian paling atas, [download latest snapshot](#) atau `git clone`.

Extract (jika download snapshot), lalu pada direktori ***di atas*** `src` (yang terdapat `Makefile`), jalankan `make`. Setelah kompilasi selesai, jalankan `./ttt`, program tersebut akan memberikan instruksi penggunaan. Jalankan server dan client di terminal session yang berbeda.

Contoh:

Menjalankan server berupa pemain manusia, dan client berupa pemain cpu

Di terminal session 1: jalankan `./ttt --server`

Di terminal session 2: jalankan `./ttt --client --pc`

TODO

- Penanganan `SIGINT` yang otomatis meng-unmap memory-mapped object