#include <sys/mman.h>

#include <sys/stat.h>

#include <errno.h>

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <fcntl.h>

#include <string.h>

struct packet{ // a packet size 100byte

int id;

short datashort[5];

long datalong[5];

double datadouble[5];

char databyte[6];

int i;

};

//char 1bytes //short 2byte //int 4bytes //long 8bytes //float 4bytes //double 8bytes //long double 16bytes

void fillpacket(struct packet \*packet,int id){

int i;

packet->id = id;

for(i=0;i<5;i++){

packet->datashort[i] = 6;

packet->datalong[i] = 6;

packet->datadouble[i] = 6.0;

}

for(i=0;i<6;i++){

packet->databyte[i] = 'x';

}

packet->i = 1;

}

int main(int argc, char\* argv[]) {

struct packet \*shmaddr;

int shmid;

char \*path = "key.text";

if(argc != 2){

fprintf(stderr,"Usage : %s <shmblocks>\n",argv[0]);

exit(1);

}else{

if(atoi(argv[1]) < 5 || atoi(argv[1]) > 30){

fprintf(stderr,"Usage : %s 5< <shmblocks> <30\n",argv[0]);

exit(1);

}

if ( ( shmid=shm\_open( path,O\_CREAT|O\_RDWR, 0666) )== -1) {

perror("shm\_open");

exit(1);

}

if ( ftruncate( shmid,atoi(argv[1])\*sizeof(struct packet) )== -1) {

perror("ftruncate");

exit(1);

}

if((shmaddr = mmap(NULL,atoi(argv[1])\*sizeof(struct packet),PROT\_READ | PROT\_WRITE,MAP\_SHARED ,shmid,0)) == MAP\_FAILED){

perror("mmap");

exit(1);

}

/\* write \*/

sleep(15);

for(int i = 0;i<5000;i++){

fillpacket(&shmaddr[i % atoi(argv[1])],i);

}

if(munmap( shmaddr,atoi(argv[1])\*sizeof(struct packet) ) == -1){

perror("munmap");

exit(1);

};

shm\_unlink("key.text");

return 0;

}

}