





Q1

(1).

a write to stdout

before fork

pid = 1001, glob = 89, var = 7

pid = 1000, glob = 88, var = 6

(2).

a write to stdout

Q2

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/stat.h>

#define BUFSIZE 4096

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

char buf[BUFSIZE];

int n, in, out;

struct stat stat\_in;

struct timespec times[2];

if (argc != 3) {

fprintf(stderr, "usage: %s srcfile newfile\n", argv[0]);

return 1;

}

if ((in = open(argv[1], O\_RDONLY)) < 0) {

fprintf(stderr, "error: fail to open %s\n", argv[1]);

return 1;

}

if ((out = open(argv[2], O\_CREAT | O\_WRONLY | O\_TRUNC,

S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IROTH)) < 0) {

fprintf(stderr, "error: fail to open %s\n", argv[2]);

return 1;

}

while ((n = read(in, buf, BUFSIZE)) > 0)

if (write(out, buf, n) != n) {

fprintf(stderr, "error: write error\n");

return 1;

}

if (fstat(in, &stat\_in) < 0) {

fprintf(stderr, "error: fail to access %s\n", argv[1]);

return 1;

}

times[0] = stat\_in.st\_atim;

times[1] = stat\_in.st\_mtim;

if (futimens(out, times) < 0) {

fprintf(stderr, "error: fail to change time of %s\n", argv[2]);

return 1;

}

close(in);

close(out);

return 0;

}

Q3

#include <stdio.h>

#include <time.h>

#include <unistd.h>

#include <fcntl.h>

#include <signal.h>

#include <stdlib.h>

#include <string.h>

long score = 0;

uid\_t euid, ruid;

void do\_log(int signo);

void game();

int main(void) {

euid = geteuid();

ruid = getuid();

if (signal(SIGALRM, do\_log) == SIG\_ERR) {

fprintf(stderr, "error: fail to set signal handler\n");

return 1;

}

alarm(3);

if (seteuid(ruid) < 0) {

fprintf(stderr, "error: fail to change ruid\n");

return 1;

}

while (1) {

game();

score++;

}

return 0;

}

void do\_log(int signo) {

char buf[32], str[64];

time\_t t;

struct tm \* tmp;

int fd;

time(&t);

tmp = localtime(&t);

strftime(buf, 32, "%F %r", tmp);

sprintf(str, "[%s]:%ld\n", buf, score);

if (seteuid(euid) < 0) {

fprintf(stderr, "error: fail to change euid\n");

exit(1);

}

if ((fd = open("scores.log", O\_WRONLY | O\_CREAT | O\_APPEND,

S\_IRUSR | S\_IWUSR)) < 0) {

fprintf(stderr, "error: fail to open scores.log\n");

exit(1);

}

write(fd, str, strlen(str));

close(fd);

if (seteuid(ruid) < 0) {

fprintf(stderr, "error: fail to change euid\n");

exit(1);

}

alarm(3);

}

void game() {

}

Q4

Accumulator

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

long total = 0;

void \* thr\_fn(void \* arg);

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

int nthread = 1;

long i;

pthread\_t pid;

if (argc != 1 && argc != 3) {

fprintf(stderr, "usage: %s [-q nthread]\n", argv[0]);

return 1;

}

if (argc == 3) {

if (argv[1][0] != '-' || argv[1][1] != 'q') {

fprintf(stderr, "usage: %s [-q nthread]\n", argv[0]);

return 1;

} else

nthread = atoi(argv[2]);

}

for (i = 0; i < nthread; i++)

pthread\_create(&pid, NULL, thr\_fn, (void \*)i);

pause();

return 0;

}

void \* thr\_fn(void \* arg) {

long pno = (long)arg, id, buf[2];

key\_t key\_base = 0x100;

int n;

if ((id = msgget(key\_base + pno, IPC\_CREAT | 0600)) < 0) {

fprintf(stderr, "error: fail to create MQ\n");

exit(1);

}

printf("queue ID %ld is %ld\n", pno, id);

while ((n = msgrcv(id, buf, sizeof(long), 1, 0)) == sizeof(long)) {

total += buf[1];

printf("sum is %ld\n", total);

}

return NULL;

}

Sendnum

#include <stdio.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

struct msg {

long mtype;

long value;

};

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

key\_t key;

long value;

int id;

struct msg m;

if (argc != 3) {

fprintf(stderr, "usage: %s key value\n", argv[0]);

return 1;

}

sscanf(argv[1], "%x", &key);

sscanf(argv[2], "%ld", &value);

if ((id = msgget(key, 0)) < 0) {

fprintf(stderr, "error: fail to get MQ\n");

return 1;

}

m.mtype = 1;

m.value = value;

if (msgsnd(id, &m, sizeof(long), 0) < 0) {

fprintf(stderr, "error: fail to send message\n");

return 1;

}

return 0;

}

Q6

static jmp\_buf env\_alrm;

改为

static sigjmp\_buf env\_alrm;

if (setjmp(env\_alrm) != 0) {

改为

if (sigsetjmp(env\_alrm, 1) != 0) {

longjmp(env\_alrm, 1);

改为

siglongjmp(env\_alrm, 1);