match.c

・２つの対戦プログラムをそれぞれ子プロセスとして起動.
・子プロセスは賭金を標準出力に書き出すので，それをパイプから読み出す.
・相手の賭金をパイプに書き出す．子プロセスはそれを標準入力から読む.
・システムコール dup2 を用いてパイプと標準入出力の入れ替えを行っている.
・子プロセスが異常終了したり，ハングアップしたときのために，シグナルを受け取ったら，子プロセスを kill する.
・これをしないと，子プロセスがゾンビとなって永久に動き続ける.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>

#define BUFSIZE 256

int sum1=0,sum2=0;
int point1=0,point2=0;
int p_to_c1[2],c1_to_p[2];
int p_to_c2[2],c2_to_p[2];
char buf[BUFSIZE];
int win1=0,win2=0;
int pid1=-1,pid2=-1;

void exit_program(int code){
  if(pid1>=0)
    kill(pid1,SIGKILL);
  if(pid2>=0)
    kill(pid2,SIGKILL);
  exit(code);
}

void sig_receive(){
  exit_program(EXIT_FAILURE);
}

void write_int(int fd, int n){
  snprintf(buf,BUFSIZE,"%d\n",n);
  write(fd,buf,strlen(buf));
  fsync(fd);
}

int read_int(int fd){
  int i;
  for(i=0;i<BUFSIZE;++i){
    read(fd,buf+i,1);
    if(buf[i]=='\n'){
      buf[i]=0;
      break;
    }
  }
  return(atoi(buf));
}

int main(int argc,char **argv){
  int i,j;
  int num_game;

  if(argc<=3){
    fprintf(stderr,"Usage: match program1 program2 num_game\n");
    exit(EXIT_SUCCESS);
  }

  num_game = atoi(argv[3]);
  if(num_game <= 0){
    fprintf(stderr,"num_game must be more than 0\n");
    exit(EXIT_FAILURE);
  }

  pipe(p_to_c1);
  pipe(c1_to_p);
  pipe(p_to_c2);
  pipe(c2_to_p);
```

```
if((pid1=fork())==0){  // child1
  close(p_to_c1[1]);
  close(c1_to_p[0]);
  dup2(p_to_c1[0],0);
  dup2(c1_to_p[1],1);
  execl(argv[1],argv[1],argv[3],NULL);
  fprintf(stderr,"Cannot execute %s\n",argv[1]);
  exit_program(EXIT_FAILURE);
}

if((pid2=fork())==0){  // child2
  close(p_to_c2[1]);
  close(c2_to_p[0]);
  dup2(p_to_c2[0],0);
  dup2(c2_to_p[1],1);
  execl(argv[2],argv[2],argv[3],NULL);
  fprintf(stderr,"Cannot execute %s\n",argv[2]);
  exit_program(EXIT_FAILURE);
}

signal(SIGHUP,sig_receive);
signal(SIGINT,sig_receive);

printf("%s vs %s\n",argv[1],argv[2]);

for(j=1;j<=num_game;++j){
  sum1=0;
  sum2=0;
  point1=0;
  point2=0;

  printf("-----------------------------------\n");
  printf("game %d\n",j);
  for(i=1;i<=10;++i){
    int bet1,bet2;
    bet1=read_int(c1_to_p[0]);
    bet2=read_int(c2_to_p[0]);
    sum1+=bet1;
    sum2+=bet2;

    if(bet1<bet2){
      point1+=bet1+bet2;
      printf("%2d: [%4d]  %4d  -> %5d %5d\n",i,bet1,bet2,point1,point2);
    } else if(bet1>bet2){
      point2+=bet1+bet2;
      printf("%2d:  %4d  [%4d] -> %5d %5d\n",i,bet1,bet2,point1,point2);
    } else {
     point1+=bet1;
     point2+=bet2;
      printf("%2d:  %4d   %4d  -> %5d %5d\n",i,bet1,bet2,point1,point2);
    }

    if(bet1<0){
      printf("%2d: %s bet %d.\n",i,argv[1],bet1);
      exit_program(EXIT_FAILURE);
    }
    if(bet2<0){
      printf("%2d: %s bet %d.\n",i,argv[2],bet2);
     exit_program(EXIT_FAILURE);
    }

    write_int(p_to_c1[1],bet2);
    write_int(p_to_c2[1],bet1);
  }

  if(sum1!=10000){
    printf("%s bet totally %d.\n",argv[1],sum1);
    exit_program(EXIT_FAILURE);
  }
  if(sum2!=10000){
    printf("%s bet totally %d.\n",argv[2],sum2);
    exit_program(EXIT_FAILURE);
  }
  if(point1>point2){
    printf("  [%5d]  %5d\n",point1,point2);
    printf("%s won.\n",argv[1]);
    win1++;
  } else if(point1<point2){
    printf("   %5d  [%5d]\n",point1,point2);
    printf("%s won.\n",argv[2]);
    win2++;
```

```
    } else {
      printf("   %5d   %5d\n",point1,point2);
      printf("Draw.\n");
    }
  }
  printf("-----------------------------------\n");
  printf("%s won : %d, %s won : %d, Draw : %d\n",argv[1], win1, argv[2], win2, num_game-win1-win2);
  exit_program(EXIT_SUCCESS);
  exit(0);
}
```