#include<stdio.h>

#include<stdbool.h>

#include<assert.h>

#include<stdlib.h>

#include<signal.h>

#include<sys/stat.h>

#include<fcntl.h>

#include"xerrors.h"

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

{

if(argc!=4)

{

printf("Uso: %s fileIn fileOut numProcess\n",argv[0]);

exit(0);

}

int numProcess=atoi(argv[3]);

int fileIn=open(argv[1],O\_RDONLY,0600);

int fileOut=open(argv[2],O\_WRONLY|O\_CREAT|O\_TRUNC,0600);

pid\_t pid;

int up[2];

xpipe(up,\_\_LINE\_\_,\_\_FILE\_\_);

for(int i=0;i<numProcess;i++)

{

pid=xfork(\_\_LINE\_\_,\_\_FILE\_\_);

if(pid==0)

{

bool end=false;

while(!end)

{

int num=-1;

close(up[1]);

int e=read(up[0],&num,sizeof(int));

assert(e!=-1);

if(e==0) end=true;

else

{

int div=0;

for(int i=1;i<=num;i++)

if(num%i==0) div++;

e=write(fileOut,&num,sizeof(int));

assert(e!=-1);

e=write(fileOut,&div,sizeof(int));

assert(e!=-1);

}

}

close(up[0]);

exit(0);

}

}

bool end=false;

while(!end)

{

int num=-1;

int e=read(fileIn,&num,sizeof(int));

assert(e!=-1);

if(e==0) end=true;

else

{

close(up[0]);

e=write(up[1],&num,sizeof(int));

assert(e!=-1);

}

}

close(up[1]);

close(fileIn);

for(int i=0;i<numProcess;i++)

{

pid=xwait(NULL,\_\_LINE\_\_,\_\_FILE\_\_);

printf("Processo %d terminato\n",getpid());

}

close(fileOut);

return 0;

}