<https://github.com/udayrg1/shell.git>

ghp\_Z4nyHKpLwUWUJiRpgHAAfst59bqkD54axDvR

PROJECT MINI\_LINUX

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include<stdlib.h>

#include<fcntl.h>

int process\_str(char \*str,char arglist[100][100],int \*nargs)

{

int i=0,j=0,k=0;

while(str[i]!='\0')

{

// printf("Enter while,arglist par i=%d,j=%d,k=%d\n",i,j,k);

if (str[i]=='\n')

{

i++;

continue;

}

else if(str[i]!= ' '){

arglist[j][k++]=str[i++];

// printf("Arglist value= %s\n",arglist[j][k]);

// printf("Enter while,arglist par i=%d,j=%d,k=%d\n",i,j,k);

}

else

{

(\*nargs)++;

arglist[j][k]='\0';

j++;

k=0;

i++;

}

}

return 0;

}

void endprogram()

{

exit(0);

}

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

{

int n,argno=0,i=0,fd1,fd2,pos;

char ch;

char path[100],buf[100],path1[100],path2[100];

char str[100], arglist[100][100];

printf("WELCOME TO MINI-LINUX\n");

// strcpy(path,"./");

while(1)

{

strcpy(path,"./");

argno=0;

fflush(stdin);

n= read(0,str,100);

// n=fgets(str,100,stdin);

str[n]='\0';

process\_str(str,arglist,&argno);

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

{

printf("Arguments passed = '%s' \n",arglist[i]);

}

if(strcmp(arglist[0],"mycp")==0)

{

printf("This is my version of cp command\n");

strcat(path,arglist[i]);

fd1 = open(arglist[1],O\_RDONLY);

if(fd1<0)

{

printf("Open Error\n");

exit(1);

}

strcpy(path,"./");

strcat(path,arglist[2]);

printf("dest path = %s\n",path);

fd2= open(arglist[2],O\_RDWR|O\_CREAT, 0666);

if(fd2<0)

{

printf("Open Error\n");

exit(1);

}

while((n=read(fd1,buf,100))>0)

{

write(fd2,buf,n);

}

close(fd1);

close(fd2);

}

else if(strcmp(arglist[0],"mymv")==0)

{

strcpy(path1,"./");

strcpy(path2,"./");

strcat(path1,arglist[1]);

strcat(path2,arglist[2]);

rename(path1,path2);

}

else if(strcmp(arglist[0],"mycat")==0)

{

printf("This is my version of cat command\n");

fd1 = open(arglist[1],O\_RDONLY);

while(1)

{

n = read(fd1,&ch,1);

if (n==0)

break;

else

printf("%c",ch);

}

close(fd1);

}

else if(strcmp(arglist[0],"myrm")==0)

{

printf("This is my version of rm command\n");

fd1=open(arglist[1],O\_RDONLY);

if(fd1==0)

{

printf("Source File not found, Copy noy possible\n");

exit(0);

}

close(fd1);

unlink(arglist[1]);

}

else if(strcmp(arglist[0],"mytail")==0)

{

FILE \* fp;

fp = fopen(arglist[1],"r");

if( fp == NULL )

{

printf("\n%s file can not be opened !!!\n",arglist[1]);

return 1;

}

char message[10][511],buffer[511];

// int i=0;

while(fgets(buffer,511,fp)){

strcpy(message[i++],buffer);

if (i>9) i=0;

}

fclose(fp);

int n=0;

while(n<10)

printf("%s",message[(i+n++)%10]);

}

else if(strcmp(arglist[0],"myhead")==0)

{

FILE \* fp;

char \*line = NULL;

size\_t len = 0;

int cnt = 0;

fp = fopen(arglist[1],"r");

// checking for file is exist or not

if( fp == NULL )

{

printf("\n%s file can not be opened !!!\n",arglist[1]);

return 1;

}

while (getline(&line, &len, fp) != -1)

{

cnt++;

if ( cnt > 10 )

break;

printf("%s",line);

fflush(stdout);

}

fclose(fp);

}

else if(strcmp(arglist[0],"exit")==0)

{

printf("Exiting from the program\n");

endprogram();

return 0;

}

}

}