#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <unistd.h> // for usleep() on Unix-like systems

#ifdef \_WIN32

#include <windows.h> // for Sleep() on Windows

#define usleep(x) Sleep(x / 1000)

#endif

#define WIDTH 80

#define HEIGHT 24

#define DROP\_SPEED 100000 // Microseconds (100000 us = 0.1 sec)

#define CHARACTERS "0123456789ABCDEF" // Limited charset for a more "hacky" look

void clear\_screen() {

// ANSI escape code to clear the screen and move the cursor to the top-left

printf("\033[H\033[J");

}

void move\_code\_rain(char screen[HEIGHT][WIDTH], int \*drops) {

for (int i = HEIGHT - 1; i > 0; i--) {

memcpy(screen[i], screen[i - 1], WIDTH \* sizeof(char));

}

// Clear the top row with spaces

memset(screen[0], ' ', WIDTH);

// Update the drop positions and generate new characters

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

int pos = drops[i] % WIDTH;

if (screen[i][pos] == ' ') {

screen[i][pos] = CHARACTERS[rand() % strlen(CHARACTERS)];

}

drops[i] += 1 + (rand() % 3); // Slightly randomize drop speed for effect

}

}

int main() {

srand(time(NULL));

char screen[HEIGHT][WIDTH];

int drops[HEIGHT];

// Initialize the screen and drop positions

clear\_screen();

memset(screen, ' ', sizeof(screen));

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

drops[i] = rand() % WIDTH;

}

printf("\033[30;41m"); // ANSI escape code to set text color to green on black background

while (1) {

// Print the current screen

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

printf("%.\*s\n", WIDTH, screen[i]);

}

// Move the code down and update the screen

move\_code\_rain(screen, drops);

// Clear the screen before the next frame

// Move the code down and update the screen

move\_code\_rain(screen, drops);

// Clear the screen before the next frame

clear\_screen();

// Sleep for a short period to control the speed of the animation

usleep(DROP\_SPEED);

}

// Reset terminal color (optional, in case the program is interrupted)

printf("\033[0m");

return 0;

}

-- 插入 -- 86,2 底端