Nízkoúrovňové programování

Pokročilá práce s terminálem

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

Nastavení vlastností terminálu

```
#include <sys/ioctl.h>
int ioctl(int fildes, int request, ... /* arg */);
```

- ovládání seriové linky (resp. znakových zařízení)
- příkaz obsahující další příkazy
- nastavení vlastností vstupu (opakování výstupu, konverze signálů, zakončení řádku)
- zjištění stavu (např. velikost terminálu)

```
struct termios ctrl;
ioctl(STDIN_FILENO, TCGETS, &ctrl);
tcflag_t old = ctrl.c_lflag;
ctrl.c_lflag = ISIG;
ioctl(STDIN_FILENO, TCSETS, &ctrl);
```

- podrobnosti, viz `man termios`

Ovládání výstupu na terminál

- emulace hardwarových terminálů
- komplexní funkcionalita
- k dispozci přes escape sekvenci `\033[`

Příklady

- `\033[2J` // vymazání terminálu
- `\033[r;cH"` // přesun na souřadnice r (row) a c (column)
- `\033[93;44m` // změna barvy písma
- více např. `https://en.wikipedia.org/wiki/ANSI_escape_code`

Systémové volání `poll` (1/2)

```
#include <poll.h>
int poll(struct pollfd fds[], nfds_t nfds, int timeout);
```

- umožňuje pasivně čekat na události spojené se soubory (proudy dat)
- test zda jsou na vstupu/výstupu data, zda došlo k zavěšení (HUP), apod.
- podrobnosti, viz `man 3 poll`

Systémové volání poll (2/2)

```
struct pollfd {
    int    fd;      // deskriptor souboru
    short  events;  // priznaky udalosti, na ktere cekame
    short  revents; //udalosti, ktere nastaly
};

struct pollfd pfd;
pfd.fd = STDIN_FILENO;  // prace se std. vstupem
pfd.events = POLLIN;    // cekame na vstup
int state = poll(&pfd, 1, timeout);
if (state < 0) return -1;
if (pfd.revents & POLLIN) { /*  data jsou na vstupu */ }
```

Hodnocené úkoly

- Nastudujte si escape sekvence a udělejte hru barevnou.
- (Volitelně) Hru vylepšete, např. počítání score, lepší grafika, NPC, pohyb v obou osách.
- Přepište v přiložené hře (alespoň dvě vámi zvolené) nízkoúrovňové funkce pro práci s terminálem do assembleru tak, aby používaly volání jádra OS.
- (Volitelně) Přepište celou hru do assembleru.