Razhroščevalnik

Seminarska naloga pri predmetu Sistemska programska oprema
Mentor: doc. Tomaž Dobravec

Jan Mrak
Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

14. januar 2025

Povzetek

Predstavitev implementacije in delovanja rezhroščevalnika na različnih platformah in za različne jezike.

1 Uvod

Razhroščevalnik je program, ki nam omogoča testiranje in upravljanje nekega drugega programa. Omogoča nam branje in pisanje spomina in registrov, poljubno ustavljanje programa, izvajanje po vrsticah ali ukazih in drugo. Poznamo različne razhroščevalnike, kot so GDB, LLDB, x64dbg, ki so bolj generalni. Obstajajo pa tudi drugačni razhroščecalniki, kot je Valgrind, ki nam omogoča pregled nad pomnilnikom procesa in recimo zaznava uhajanje spomina (angl. memory leak).

2 Razhroščevalnik na sistemu Linux

Na Unix in Unix-like sistemih je ponujen sistemski klic

```
long ptrace(enum __ptrace_request op, pid_t pid, void *addr, void *data);
```

, ki nam omogoča, da lahko dostopamo do prcesa podanega s `pid`. Da pa bo sistemski klic uspel, mora proces, do katerega želimo dostopati, dovoliti dostop do njega. To pa lahko naredi s klicem sistemske funkcije `ptrace`:

```
// pid, addr in data argumenti so ignorirani
ptrace(PTRACE_TRACEME, 0, NULL, NULL);
```

Po tem bo razhroščevalnik lahko dostopal do tega procesa.

Razhroščevalnik lahko sam ustvari proces, ki potem pokliče `ptrace` z argumentom `PTRACE_TRACEME`,

```
int pid = fork();
if (pid == 0) {
    ptrace(PTRACE_TRACEME, 0, NULL, NULL);
    execve(...);
}
```

lahko pa se priklopi na nek obstoječi proces z uporabo `PTRACE_ATTACH`, ki pošlje signal `SIGSTOP`, da se proces ustavi, ali pa `PTRACE_SEIZE`, ki ne ustavi procesa.

```
ptrace(PTRACE_ATTACH, pid, NULL, NULL);
// ali
ptrace(PTRACE_SEIZE, pid, NULL, PTRACE_O_flags);
```

Če želimo ustaviti proces, ga lahko ustavimo kadarkoli s klicem `ptrace` in za argument `op` izberemo `PTRACE_INTERRUPT`. Ko pa proces ustavimo, imamo na voljo veliko različnih možnosti za upravljanje s procesom.

Možnosti za pridobivanje in upravljanje z informacijami:

- `PTRACE_PEEKDATA` ali `PTRACE_PEEKTEXT`, ki nam omogočata, da beremo iz procesovega spomina
- `PTRACE_POKEDATA` ali `PTRACE_POKETEXT`, ki nam omogočata, da pišemo v spomin procesa
- `PTRACE_GETREGS` ali `PTRACE_GETFREGS`, ki nam omogočata, da preberemo splošno namenske registre ali registre za delanje s plavajočo vejico
- `PTRACE_SETREGS` ali `PTRACE_SETFREGS`, podobno kot pri prejšnjem primeru, dobimo dostop do registrov in v njih lahko zapišemo vrednosti
- `PTRACE_GETSIGINFO`, ki pridobi informacije o signalu, ki je ustavil proces
- `PTRACE_PEEKSIGINFO`, enako pridobi informacije o signalu, vendar ga ne vzame iz vrste signalov

Možnosti za upravljanje poteka procesa:

- `PTRACE_CONT`, ki znova zažene ustavljen proces, da nadaljuje z delovanjem
- `PTRACE_SINGLESTEP`, ki izvede le en ukaz
- `PTRACE_SYSCALL`, ki se vede kot `PTRACE_CONT`, vendar se preces, ki ga razhroščijemo ustavi tik pred vstopom v sistemski klic, oziroma ob izstopu sistemskega klica
- `PTRACE_KILL`, ki procesu pošlje signal `SIGKILL` in ga tako prisilno zaključi
- `PTRACE_INTERRUPT`, ki ustavi proces

Obstaja še veliko drugih možnosti za delo s procesom, ki pa so razložene v priročniku man za ptrace [1].

2.1 DWARF format

3 Razhroščevalnik na sistemu Windows

4 Razhroščevalnik za jezik Java / Python ?

Literatura

[1] ptrace(2) - linux manual page. [Online] Dosegljivo: `https://man7.org/linux/man-pages/man2/ptrace.2.html`. Zadnji obisk 14. 1. 2025.