

Implementation of a debugger under Linux (on x86 machines, of course :)

Stephane A. Sezer

December 8, 2011

- 1 Debugging a program on an Intel CPU
 - Interruptions
 - Generating an interrupt
 - Debug registers

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

Interruptions

Generating an interrupt

Debug registers

`ptrace(2)`

Questions ?

1 Debugging a program on an Intel CPU

- Interruptions
- Generating an interrupt
- Debug registers

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

Interruptions

Generating an interrupt

Debug registers

ptrace(2)

Questions ?

- Redirection of the control flow to the kernel
- The kernel handle the interrupts and gives back the control to userland
- Used for debug traps

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

Interruptions

Generating an interrupt
Debug registers

ptrace(2)

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

1 Debugging a program on an Intel CPU

- Interruptions
- **Generating an interrupt**
- Debug registers

Debugging a
program on an
Intel CPU

Interruptions

Generating an interrupt

Debug registers

ptrace(2)

Questions ?

- Stop execution of the program and generate an interrupt
- `int3 / 0xCC`
- `int $0x03 / 0xCD03`
- Used to implement “regular” breakpoints in GDB

1 Debugging a program on an Intel CPU

- Interruptions
- Generating an interrupt
- Debug registers

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

Interruptions

Generating an interrupt

Debug registers

ptrace(2)

Questions ?

- DR0 - DR7
- Used to implement GDB's `watch` and `hbreak` commands
- 3v1l ;)

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

Interruptions

Generating an interrupt

Debug registers

`ptrace(2)`

Questions ?

- 2 ptrace(2)
 - Working with memory
 - Working with registers
 - Playing with the process' control flow
 - Sending signals to the process

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

- Working with memory
- Working with registers
- Playing with the process' control flow
- Sending signals to the process

Questions ?

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

- 2 ptrace(2)
 - Working with memory
 - Working with registers
 - Playing with the process' control flow
 - Sending signals to the process

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

- `PTRACE_PEEKDATA`, `PTRACE_POKEDATA`
- `PTRACE_PEEKTEXT`, `PTRACE_POKETEXT`
- read/write to/from the process' memory
- `ptrace(PTRACE_PEEKDATA, pid, 0x1337, NULL);`

- 2 ptrace(2)
 - Working with memory
 - Working with registers
 - Playing with the process' control flow
 - Sending signals to the process

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

- `struct user;`
- `struct user_regs_struct;`
- `PTRACE_PEEKREGS, PTRACE_POKEREGS`
- `read/write to/from the process' register base`

struct user_regs_struct

```
struct user_regs_struct
{
    long int ebx;
    long int ecx;
    long int edx;
    long int esi;
    [...]
    long int eip;
    long int xcs;
    long int eflags;
    long int esp;
    long int xss;
};
```

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

`ptrace(2)`

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

- 2 ptrace(2)
 - Working with memory
 - Working with registers
 - **Playing with the process' control flow**
 - Sending signals to the process

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

**Playing with the process'
control flow**

Sending signals to the
process

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

`ptrace(2)`

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

- `PTRACE_CONT`, to continue execution
- `PTRACE_SYSCALL`, to continue execution until next syscall

- 2 ptrace(2)
 - Working with memory
 - Working with registers
 - Playing with the process' control flow
 - Sending signals to the process

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

`ptrace(2)`

Working with memory

Working with registers

Playing with the process'
control flow

Sending signals to the
process

Questions ?

- `PTRACE_CONT` can take an additional argument
- Used to implement GDB's `signal` command

Questions ?

Implementation of
a debugger under
Linux (on x86
machines, of
course :)

Stephane A. Sezer

Debugging a
program on an
Intel CPU

ptrace(2)

Questions ?

WUT?