

Hello World

Mitsuru Takigahira

自己紹介

名前 瀧ヶ平 充

どんな人 研修でだいぶ察した方も多いと思います

今日は

Hello World\n の話をします

Hello World といえば

```
#include <stdio.h>
int main(void) {
    printf("Hello World\n");
    return 0;
}
```

Hello World

違います

甘えるな

```
#include <stdio.h>
```

なんですかこれは、甘えるのも大概にしないで。

おまじないをやめろ

おまじないで使える printf くんの仕事

- ① 第一引数のフォーマット解析、文字列の作成
- ② write システムコールを呼び、標準出力に出力

Hello World くらいなら

printf なんかに頼らなくてもできる！

候補 1

```
#include <unistd.h>
int main(void) {
    char *str = "Hello World\n";
    write(1, str, 12);
}
```

だから甘えるな

```
#include <unistd.h>
```

なんちゃら.h を使わずにやるには

- `%eax` のレジスタに対応するシステムコールの番号を入れる
- 引数に対応するレジスタに適切な値を入れる
- `syscall` 命令を実行する

つまり

gcc のインラインアセンブラを使おう

インラインアセンブラとは

C のコードの中にアセンブラが書ける → うれしい！
C の変数をレジスタに入れられる → うれしい！

まず `man syscall` と `man 2 write` を見ましょう

- x86_64 の場合、引数ははじめから順に `%rdi`, `%rsi`, `%rdx` ...
- `write` システムコールの番号は 1
- `write` システムコールの引数は
 - ファイル記述子
 - 出力する文字列の最初の値のポインタ
 - 何バイト出力するか

結論

```
int main (void) {  
    char *str = "hello world\n";  
    __asm__("mov %0, %%rsi;"  
        :  
        : "r"(str)  
        : "%rsi");  
    __asm__("mov $1, %rdi;"  
        "mov $1, %eax;"  
        "mov $12, %rdx;"  
        "syscall");  
    return 0;  
}
```

こういう話やもっとやばい発表が聞ける kernel/vm 探検隊@東京
7/21 に IIJ で開催されます。興味があればご参加ください。