

# tod – the TOY disassembler

B96902122 飯田來夢

- 研究動機

這門課讓我了解 CPU 的基本機構和機械語言與組合語言的關聯。經過逐步的修習，我感覺到對 TOY machine 的了解有助於組合語言的學習。因此我想更深入地理解組合語言，趁著這個機會寫一種編譯程式，即 TOY disassembler。

- 功能

這是 tod 的 Readme file.

1. How to use tod

To disassemble a single TOY toy file 'file.toy'

type the following:

```
tod file.toy
```

This command creates file 'file.asmd' in the current directory.

If such file already exists tod overwrites it.

If you want to specify your own output file name, type the following:

```
tod -r file.asm file.toy
```

2. The command-line options

-r <output filename> - rename output file. Default name is '(input file name except extension).asmd'.

Ex.

```
>tod file.toy -r file2.disasm file2.toy file3.toy
```

```
output: 'file.asmd' <- input: 'file.toy'
```

```
output: 'file2.disasm' <- input: 'file2.toy'
```

```
tod: output file name of 'file3.toy' will be changed to default name
```

```
output: 'file3.asmd' <- input: 'file3.toy'
```

-s - simulates toy machine and decides whether the code is variable. If the input file does not satisfy condition 1 of the convention, please try this option.

-n - no dup. If you don't want to recover DUP directive, please use this option.

### 3. Convention

tod expects input file satisfies following conditions.

condition 1 : Variables appear before instructions.

condition 2 : First instruction is in the starting address 0x10. (tod supports "some tricks" of toyasm. If the code in 0x10 is 'C0\*\*', then tod ignores this instruction)

condition 3 : Instructions do not appear before 0x10.