

Preccs: 実用的な通信プロトコルコンパイラ

服部健太 (株式会社 システム計画研究所)

簡単な仕様記述から通信プロトコルの処理を行うC言語のコードを自動的に生成する

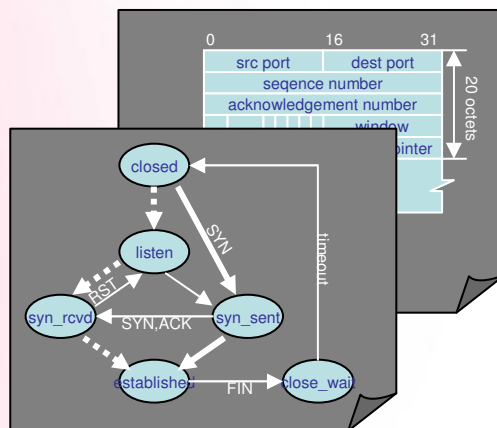
通信プロトコルコンパイラ **Preccs** を開発

独自の最適化技術により生成コードの処理性能が大幅に向上

複雑な通信プロトコルの仕様も

簡単に記述

プログラムも自動生成



```
/* オプション形式の定義 */
type Option = {{
    tag : octet;
    len : octet;
    data : octet[len]
}}

/* メインプロセス定義 */
proc Main() = Init()
/* 応答待ちプロセス定義 */
proc WaitReply(sPkt:DhcpReq) =
    timer!10 -> Init()
    | sockin?msg:octet* ->
        ( msg @ DhcpAck -> Bound(sPkt)
          | DhcpNak -> Init()
          | _ -> Error(msg);
          Init() )
```

Preccs
コンパイラ

```
/* extended NFA */
nfa_t __nfa_table[] = {
    /* [0] */ { NFA_MATCH, 2, 0, 1, -1 },
    /* [1] */ { NFA_EPSLN, 0, 0, 2, -1 },
    /* [2] */ { NFA_GETVS, 0, 0, 7, -1 },
    /* [3] */ { NFA_PSHVS, 0, 0, 0, -1 },
    /* static data */
    char_data_8CF9F8[] = "x01";
    value_blk_t_val_8cf9e0 = { VTYPE_BLK, 0, 1, _data_8CF9F8 };
    char_data_8CFA20[] = "x01";
    value_blk_t_val_8cfa08 = { VTYPE_BLK, 0, 1, _data_8CFA20 };
    char_data_8CFA40[] = "x06";
    /* state machine */
    void disp_proc(proc_t "proc") {
        next:
        switch (proc->state) {
            case 0: /* InitProc */
                {
                    proc_t "np" = make_proc(1, 0);
                    proc_setrdy(np, 1);
                }
                free_proc(proc); break;
            case 1: /* Main */
                {
                    proc_t "np" = make_proc(2, 1);
                    proc_setrdy(np, 2);
                }
                free_proc(proc); break;
            case 2: /* Dhcpinit */
                {
                    proc->data[0] = (void *)&_val_8cfa08;
                }
                break;
        }
    }
}
```

Preccsを使えば...

- 複雑な非同期IO処理を直感的に記述
- 実装時のつまらないバグを撲滅
- 生産性の大幅な向上

Preccsを用いて実装したhttpサーバを実装はApacheと同等以上の処理性能を発揮
VoIPアプリケーションもわずか300行程度
(CODEC除く)で記述できた

公式サイト <http://preccs.isp.jp>