

## ソケットを使ってパケットのやりとりを行う

```
(define-module gsup.net.packetutil
  (use srfi-1)
  (use gauche.net)
  (use binary.pack)
  (export-all)
)

(select-module gsup.net.packetutil)

(define (send-packet-with-socket socket packet)
  (let* ((fd (socket-fd socket))
         (out-port (open-output-fd-port fd)))
    (for-each (lambda (x) (write-byte x out-port)) packet)
    (flush out-port)))

(define (send-udp-packet-with-addr src dest port packet)
  (let ((socket (make-socket AF_INET SOCK_DGRAM)))
    (socket-bind socket (make <sockaddr-in> :host src :port port))
    (socket-connect socket (make <sockaddr-in> :host dest :port port))
    (send-packet-with-socket socket packet))
  ; (socket-close socket))

(define (recv-packet-with-socket socket length)
  (let* ((fd (socket-fd socket))
         (in-port (open-input-fd-port fd)))
    (unpack "C*" :from-string (read-block length in-port))))

(define (recv-udp-packet-with-addr src dest port length)
  (let ((socket (make-socket AF_INET SOCK_DGRAM)))
    (socket-bind socket (make <sockaddr-in> :host src :port port))
    (socket-connect socket (make <sockaddr-in> :host dest :port port))
    (recv-packet-with-socket socket length))
  ; (socket-close socket))

(define (sendrecv-udp-packet-with-addr src dest port packet length)
  (let ((socket (make-socket AF_INET SOCK_DGRAM)))
    (socket-bind socket (make <sockaddr-in> :host src :port port))
    (socket-connect socket (make <sockaddr-in> :host dest :port port))
    (send-packet-with-socket socket packet)
    (recv-packet-with-socket socket length))
  ; (socket-close socket))

(define (sendrecv-packet-with-socket socket packet length)
  (send-packet-with-socket socket packet)
  (recv-packet-with-socket socket length))

(define (sendrecv-udp-packet-with-addr src dest port packet length)
  (let ((socket (make-socket AF_INET SOCK_DGRAM)))
    (socket-bind socket (make <sockaddr-in> :host src :port port))
    (socket-connect socket (make <sockaddr-in> :host dest :port port))
    (send-packet-with-socket socket packet)
    (recv-packet-with-socket socket length))

(define (hton32 value)
  (list
   (bit-field value 24 32)
   (bit-field value 16 23)
   (bit-field value 8 15)
   (bit-field value 0 7)))

(define (hton16 value)
  (list
   (bit-field value 8 15)
   (bit-field value 0 7)))

(define (ntoh32 vlist)
  (if (not (= (length vlist) 4))
      #f
      (+ (ash (car vlist) 24)
```

```
(ash      (car (cdr vlist)) 16)
(ash     (car (cdr (cdr vlist))) 8)
(ash (car (cdr (cdr (cdr vlist)))) 0)))

(define (ntoh16 vlist)
  (if (not (= (length vlist) 2))
      #f
      (+ (ash      (car (cdr (cdr vlist))) 8)
         (ash (car (cdr (cdr (cdr vlist)))) 0))))))

(provide "gsup.net.packetutil")
```