

```

>> (define fac
      (lambda (n)
        (if (= n 0)
            1
            (* n (fac (- n 1))))))
| (make-closure
   (n)
   (if-exp
    (app-exp (var-exp =) ((var-exp n) (lit-exp 0)))
    (lit-exp 1)
    (app-exp
     (var-exp *)
     ((var-exp n)
      (app-exp
       (var-exp fac)
       ((app-exp (var-exp -) ((var-exp n) (lit-exp 1))))))))))
   ())
| (closure
   (n)
   (if-exp
    (app-exp (var-exp =) ((var-exp n) (lit-exp 0)))
    (lit-exp 1)
    (app-exp
     (var-exp *)
     ((var-exp n)
      (app-exp
       (var-exp fac)
       ((app-exp (var-exp -) ((var-exp n) (lit-exp 1))))))))))
   ())
| (extend-global-env
   fac
   (closure
    (n)
    (if-exp
     (app-exp (var-exp =) ((var-exp n) (lit-exp 0)))
     (lit-exp 1)
     (app-exp
      (var-exp *)
      ((var-exp n)
       (app-exp
        (var-exp fac)
        ((app-exp (var-exp -) ((var-exp n) (lit-exp 1))))))))))
    ()))
|#<void>
#<void>
>> (fac 2)
| (apply-env () fac)
| (apply-global-env fac)
| (closure
   (n)
   (if-exp
    (app-exp (var-exp =) ((var-exp n) (lit-exp 0)))
    (lit-exp 1)
    (app-exp
     (var-exp *)
     ((var-exp n)
      (app-exp
       (var-exp fac)
       ((app-exp (var-exp -) ((var-exp n) (lit-exp 1))))))))))
   ()))

```

```

        (var-exp *)
        ((var-exp n)
         (app-exp
          (var-exp fac)
          ((app-exp (var-exp -) ((var-exp n) (lit-exp 1)))))))
    ())
| (extend-env (n) (2) ())
| ((n) . #1(2))
| (apply-env ((n) . #1(2)) =)
| (apply-env () =)
| (apply-global-env =)
| (primitive =)
| (apply-env ((n) . #1(2)) n)
| 2
| (apply-env ((n) . #1(2)) *)
| (apply-env () *)
| (apply-global-env *)
| (primitive *)
| (apply-env ((n) . #1(2)) n)
| 2
| (apply-env ((n) . #1(2)) fac)
| (apply-env () fac)
| (apply-global-env fac)
| (closure
   (n)
   (if-exp
    (app-exp (var-exp =) ((var-exp n) (lit-exp 0)))
    (lit-exp 1)
    (app-exp
     (var-exp *)
     ((var-exp n)
      (app-exp
       (var-exp fac)
       ((app-exp (var-exp -) ((var-exp n) (lit-exp 1))))))))))
   ())
| (apply-env ((n) . #1(2)) -)
| (apply-env () -)
| (apply-global-env -)
| (primitive -)
| (apply-env ((n) . #1(2)) n)
| 2
| (extend-env (n) (1) ())
| ((n) . #1(1))
| (apply-env ((n) . #1(1)) =)
| (apply-env () =)
| (apply-global-env =)
| (primitive =)
| (apply-env ((n) . #1(1)) n)
| 1
| (apply-env ((n) . #1(1)) *)
| (apply-env () *)
| (apply-global-env *)
| (primitive *)
| (apply-env ((n) . #1(1)) n)

```

```
|1
|(apply-env ((n) . #1(1))) fac)
|(apply-env () fac)
|(apply-global-env fac)
|(closure
  (n)
  (if-exp
    (app-exp (var-exp =) ((var-exp n) (lit-exp 0)))
    (lit-exp 1)
    (app-exp
      (var-exp *)
      ((var-exp n)
        (app-exp
          (var-exp fac)
          ((app-exp (var-exp -) ((var-exp n) (lit-exp 1))))))))))
  ())
|(apply-env ((n) . #1(1)) -)
|(apply-env () -)
|(apply-global-env -)
|(primitive -)
|(apply-env ((n) . #1(1)) n)
|1
|(extend-env (n) (0) ())
|((n) . #1(0))
|(apply-env ((n) . #1(0)) =)
|(apply-env () =)
|(apply-global-env =)
|(primitive =)
|(apply-env ((n) . #1(0)) n)
|0
2
>>
```