

LOOPS provides two special versions of message sending that start a separate process to run LOOPS methods. These are `←Process` and `←Process!` which are analogous to `←` and `←!`.

`(←Process obj sel arg1 ... argn)`

[Macro]

- Purpose:** Starts a new process to run the selected method on the object, *obj*.
- Behavior:** The method indicated by *sel* is run in a separate process for the given instance or class, *obj*. See the *Interlisp-D Reference Manual* for a discussion of processes.
- Arguments:** *obj* A LOOPS object.
sel Name of the method to be executed as a process.
arg1 ... argn Arguments for the method specified in *sel*.
- Returns:** Pointer to a process data type.
- Example:** Assume the method **ClockTime** is added to the class **LCD**, as follows:

```
[Method ((LCD ClockTime
self WaitTime DisplaySeconds?)
(while T
  do (←@ self reading
      [MKATOM (DATE (if DisplaySeconds?
                    then (DATEFORMAT NO.DATE)
                    else (DATEFORMAT NO.DATE NO.SECONDS))]
      (← self Update)
      (BLOCK (OR WaitTime 1000]))
(LCD.ClockTime)
```

ClockTime takes two arguments: **WaitTime**, the wait time between updates of the **LCD** reading, and **DisplaySeconds?**, a flag used to determine if seconds are to be displayed on the **LCD**. **ClockTime** runs an infinite loop which sets the **LCD** reading, updates the **LCD** display, and blocks the **ClockTime** loop to allow other system processes to run. The command

```
(←Process ($ LCDInstance1) ClockTime 60000)
```

adds the process **ClockTime** to the process list and (**\$ LCDInstance1**) becomes a digital clock which updates itself every minute.

19:33

`(←Process! obj sel arg1 argn)`

[Macro]

- Purpose:** Starts a new process to run the selected method on the object *obj*. Like `←Process`, except the argument *sel* is evaluated.
- Behavior:** Evaluates *sel* returns a selector for a method of *obj*. This method is run on a separate process for the given instance or class, *obj*.
- Arguments:** *obj* A LOOPS object.
sel Name of the method to be executed as a process.
arg1 argn Arguments needed for the method.
- Returns:** Pointer to the process data type.
- Example:** Assume the variable **LCDClock** is set to **ClockTime**, which is the method added to the **LCD** class as described for `←Process`. The command
- ```
(←Process! ($ LCDInstance1) LCDClock 2000 T)
```
- adds the process **LCDClock** to the process list and (**\$ LCDInstance**) becomes a digital clock with a seconds display which updates itself every two seconds.

**19:49:39**