

Users' Briefing of the
Full Duplex Input PROPOSAL

Please bring any comment to the attention of the Systems Group.

While no `□RECEIVE` is directed at a particular terminal, characters keyed into that terminal are either

- 1) added to an anticipatory buffer, if there is room, without ever implying a confirming echo, or
- 2) lost, if the anticipatory buffer is full, with an optionally implied warning bell.

The anticipatory buffer can hold no more than N characters where N is zero if type-ahead is disabled.

When a process `□RECEIVE`'s from a terminal, ARS moves characters out of the anticipatory buffer and into a working buffer until

- a) an activation character is moved. This completes the `□RECEIVE` , or
- b) the `□RECEIVE` is completed by another source, such as an alarm clock, or
- c) the anticipatory buffer is empty. Here, ARS waits for the typist to continue. Subsequently keyed in characters are moved directly to the working buffer until (a) or (b) occur.

Characters moved into the working buffer are echoed unless input echoing is disabled.

`VMIX ← □MIX_OF CHAR`

performs conversion from character to mixed.

$\nabla \text{OLD_MODE} \leftarrow \square \text{TERM} \text{MODE NEW_MODE}$

sets the new terminal mode and returns the old terminal mode.

Terminal mode is a four-character vector as follows:

$\text{MODE}[1] = 'T' \Rightarrow$ typeahead;	$[1] = '~' \Rightarrow$ disable
$[2] = 'C' \Rightarrow$ conversion;	$[2] = '~' \Rightarrow$ no conversion
$[3] = 'E' \Rightarrow$ echo input;	$[3] = '~' \Rightarrow$ don't echo
$[4] = 'B' \Rightarrow$ bell if lost;	$[4] = '~' \Rightarrow$ ignore if lost
$[i] = '\underline{S}' \Rightarrow$ same as $\text{OLD_MODE}[i]$, for integer $1 \leq i \leq 4$	

Initially, $\text{MODE} \leftarrow '\sim\text{CEB}'$.

$\nabla \square \text{ACTIVATION } A$

defines the activation characters, which are characters that terminate a $\square \text{RECEIVE}$.

If A is scalar, then

$A = 0 \Rightarrow$ Everything up to, but excluding, the next carriage return is input. (Normal input.)

$A = 1 \Rightarrow$ Everything up to, and including, the next character is input. (Char-by-char input.)

Otherwise, A must be a non-empty character vector.

Everything up to, and including, any element of this vector is input.

The following primitives may not be called while type-ahead is enabled, i.e., $\text{MODE}[1] = 'T'$.

$\nabla \square \text{PROMPT VALUE}$

stores VALUE internally until the next $\square \text{RECEIVE}$ is issued to the terminal. At that time, VALUE is printed, then immediately input is accepted. This guarantees that input keyed in after the prompt will not be lost. It is an error to output to the terminal after a $\square \text{PROMPT}$ but before the next $\square \text{RECEIVE}$. .

▽ *CHARS* ← □*FLUSH*

returns the characters currently in the working buffer and then empties that buffer.

▽ □*UNFLUSH CHARS*

place *CHARS* in the working buffer. If *MODE*[3] = 'E' (echo), *CHARS* are printed. It is an error if the working buffer was not empty or if any element of *CHARS* is currently an activation character.

Note: □*UNFLUSH* □*FLUSH* re-echoes aborted input. See (b) above.