□0 ITRON		TS_DORMANT	TS_RUNNABLE	TS_WAITING	TS_SUSPENDED	TS_WAITING_SUSPENDED
Ⓣ	E╲S	0	1	2	3	4

act_tsk	0	TS_RUNNABLE
		if (make_active() == TRUE) {
		  dispatch();
		}
		return E_OK;
			actcnt == FALSE
			-
			actcnt = TRUE;
			return E_OK;
			else
			-
			return E_QOVR;
				actcnt == FALSE
				-
				actcnt = TRUE;
				return E_OK;
				else
				-
				return E_QOVR;
					actcnt == FALSE
					-
					actcnt=TRUE;
					return E_OK;
					else
					-
					return E_QOVR;
						actcnt == FALSE
						-
						actcnt = TRUE;
						return E_OK;
						else
						-
						return E_QOVR;

ext_tsk	1	×	actcnt == TRUE
			-
			make_non_runnable();
			make_dormant();
			actcnt = FALSE;
			make_active();
			tstat = TS_RUNNABLE; /* */
			else
			TS_DORMANT
			make_non_runnable();
			make_dormant();
			tstat = TS_DORMANT; /* */
				actcnt == TRUE
				TS_RUNNABLE
				make_non_runnable();
				make_dormant();
				actcnt = FALSE;
				make_active();
				tstat = TS_RUNNABLE; /* */
				else
				TS_DORMANT
				make_non_runnable();
				make_dormant();
				tstat = TS_DORMANT; /* */
					actcnt == TRUE
					TS_RUNNABLE
					make_non_runnable();
					make_dormant();
					actcnt = FALSE;
					make_active();
					tstat = TS_RUNNABLE; /* */
					else
					TS_DORMANT
					make_non_runnable();
					make_dormant();
					tstat = TS_DORMANT; /* */
						actcnt == TRUE
						TS_RUNNABLE
						make_non_runnable();
						make_dormant();
						actcnt = FALSE;
						make_active();
						tstat = TS_RUNNABLE;
						else
						TS_DORMANT
						make_non_runnable();
						make_dormant();
						tstat = TS_DORMANT; /* */

ter_tsk	2	-
		return E_OBJ;
			actcnt == TRUE
			-
			make_non_runnable();
			make_dormant();
			actcnt = FALSE;
			if (make_active() == TRUE) {
			  dispatch();
			}
			tstat = TS_RUNNABLE; /* */
			return E_OK;
			else
			TS_DORMANT
			make_non_runnable();
			make_dormant();
			tstat = TS_DORMANT; /* */
			return E_OK;
				actcnt == TRUE
				TS_RUNNABLE
				wait_cancel();
				make_dormant();
				actcnt = FALSE;
				if (make_active() == TRUE) {
				  dispatch();
				}
				tstat = TS_RUNNABLE; /* */
				return E_OK;
				else
				TS_DORMANT
				wait_cancel();
				make_dormant();
				tstat = TS_DORMANT; /* */
				return E_OK;
					actcnt == TRUE
					TS_RUNNABLE
					make_dormant();
					actcnt = FALSE;
					if (make_active() == TRUE) {
					  dispatch();
					}
					tstat = TS_RUNNABLE; /* */
					return E_OK;
					else
					TS_DORMANT
					make_dormant();
					tstat = TS_DORMANT; /* */
					return E_OK;
						actcnt == TRUE
						TS_RUNNABLE
						wait_cancel();
						make_dormant();
						actcnt = FALSE;
						if (make_active() == TRUE) {
						  dispatch();
						}
						tstat = TS_RUNNABLE; /* */
						return E_OK;
						else
						TS_DORMANT
						wait_cancel();
						make_dormant();
						tstat = TS_DORMANT; /* */
						return E_OK;

slp_tsk	3	×	wupcnt == TRUE
			-
			wupcnt = FALSE;
			return E_OK;
			else
			TS_WAITING
			tstat = TS_WAITING | TS_WAIT_SLEEP;
			make_wait();
			dispatch();
			return wercd;
				×	×	×

wup_tsk	4	-
		return E_OBJ;
			wupcnt == FALSE
			-
			wupcnt = TRUE;
			return E_OK;
			else
			-
			return E_QOVR;
				(tstat & TS_WAIT_SLEEP) != 0
				TS_RUNNABLE
				if (wait_complete() == TRUE) {
				  dispatch();
				}
				tstat = TS_RUNNABLE; /* */
				return E_OK;
				else
				wupcnt == FALSE
				-
				wupcnt = TRUE;
				return E_OK;
				else
				-
				return E_QOVR;
					wupcnt == FALSE
					-
					wupcnt = TRUE;
					return E_OK;
					else
					-
					return E_QOVR;
						(tstat & TS_WAIT_SLEEP) != 0
						TS_SUSPENDED
						wait_complete();
						tstat = TS_SUSPENDED; /* */
						return E_OK;
						else
						wupcnt == FALSE
						-
						wupcnt = TRUE;
						return E_OK;
						else
						-
						return E_QOVR;

rel_wai	5	-
		return E_OBJ;
			-
			return E_OBJ;
				TS_RUNNABLE
				if (wait_release() == TRUE) {
				  dispatch();
				}
				tstat = TS_RUNNABLE; /* */
				return E_OK;
					-
					return E_OBJ;
						TS_SUSPENDED
						wait_release();
						tstat = TS_SUSPENDED; /* */
						return E_OK;

sus_tsk	6	-
		return E_OBJ;
			TS_SUSPENDED
			tstat = TS_SUSPENDED;
			if (make_non_runnable() == TRUE) {
			  dispatch();
			}
			return E_OK;
				TS_WAITING_SUSPENDED
				tstat |= TS_SUSPENDED;
				return E_OK;
					-
					return E_QOVR;
						-
						return E_QOVR;

rsm_tsk	7	-
		return E_OBJ;
			-
			return E_OBJ;
				-
				return E_OBJ;
					TS_RUNNABLE
					if (make_runnable() == TRUE) {
					  dispatch();
					}
					tstat = TS_RUNNABLE; /* */
					return E_OK;
						TS_WAITING
						tstat &= ~TS_SUSPENDED;
						return E_OK;

tslp_tsk	8	×	wupcnt == TRUE
			-
			wupcnt = FALSE;
			return E_OK;
			else
			tmout == TMO_POL
			-
			return E_TMOUT;
			else
			TS_WAITING
			tstat = TS_WAITING | TS_WAIT_SLEEP;
			make_wait_tmout();
			dispatch();
			return wercd;
				×	×	×

dly_tsk	9	×	TS_WAITING
			tstat = TS_WAITING;
			make_non_runnable();
			tmevtb_enqueue();
			dispatch();
			return wercd;
				×	×	×