

Datorsystemteknik

för D2, ICT2, E3 och Mek3

Nicholas Wickström

Högskolan i Halmstad
Sverige

Outline

- Föregående fl.
- Testning

Skriv ett program som testar kodstycket runt /* 1 */, /* 2 */, /* 3 */ (kärnans funktioner som skall användas finns i bilaga).

<p>Function send_wait Disable interrupt Save context IF first execution THEN Set: "not first execution any more" IF receiving task is waiting THEN Copy sender's data to the data area of the receivers Message /* 1 */ Remove receiving task's Message struct from the mailbox Move receiving task to Readylist ELSE Allocate a Message structure Set data pointer /* 2 */ Add Message to the Mailbox Move sending task from Readylist to Waitinglist ENDIF Load context ELSE IF deadline is reached THEN Disable interrupt Remove send Message /* 3 */ Enable interrupt Return DEADLINE_REACHED ELSE Return OK ENDIF ENDIF</p>	<p>exception send_wait(Mailbox* mbox, void* Data)</p> <p>This call will send a Message to the specified Mailbox. If there is a receiving task waiting for a Message on the specified Mailbox, send_wait will deliver it and the receiving task will be moved to the Readylist. Otherwise, if there is not a receiving task waiting for a Message on the specified Mailbox, the sending task will be blocked. In both cases (blocked or not blocked) a new task schedule is done and possibly a context switch. During the blocking period of the task its deadline might be reached. At that point in time the blocked task will be resumed with the exception: DEADLINE_REACHED. Note: send_wait and send_no_wait Messages shall not be mixed in the same Mailbox.</p> <p>Argument *mbox a pointer to the specified Mailbox. *Data: a pointer to a memory area where the data of the communicated Message is residing.</p> <p>Return parameter exception: The exception return parameter can have two possible values:</p> <ul style="list-style-type: none"> • OK: Normal behavior, no exception occurred. • DEADLINE_REACHED: This return parameter is given if the sending tasks' deadline is reached while it is blocked by the send_wait call.
--	---

- 1) Besök alla kodstycken dvs. om man satt en brytpunkt på /*1*/, /*2*/ samt /*3*/ skall programmet stanna på just det ställena.
- 2) Testa att data kopierats som det skall.
- 3) Testa på något sätt om meddelanden är tillagda resp. borttagna ur mailboxen.

```

#define TEST_PATTERN_1 0xAA
#define TEST_PATTERN_2 0x55

mailbox *mb;

int nTest1=0, nTest2=0, nTest3=0;
void main(void)
{
    if (init_kernel() != OK ) {
        /* Memory allocation problems */
        while(1);
    }

    if (create_task( task1, 2000 ) != OK ) {
        /* Memory allocation problems */
        while(1);
    }

    if (create_task( task2, 4000 ) != OK ) {
        /* Memory allocation problems */
        while(1);
    }

    if ( (mb=create_mailbox(1,sizeof(int))
        ) == NULL) {
        /* Memory allocation problems */
        while(1);
    }

    run(); /* First in readylist is task1 */
}

```

```

void task1(void)
{
volatile int nData_t1 = TEST_PATTERN_1;
wait(10); /* task2 börjar köra */

if( no_messages(mb) != 1 )
    terminate(); /* ERROR */

if(send_wait(mb,&nData_t1) == DEADLINE_REACHED)
    terminate(); /* ERROR */

wait(10); /* task2 börjar köra */

/* start test 2 */
nData_t1 = TEST_PATTERN_2;

if(send_wait(mb,&nData_t1) == DEADLINE_REACHED)
    terminate(); /* ERROR */

wait(10); /* task2 börjar köra */

/* start test 3 */
if(send_wait(mb,&nData_t1)==DEADLINE_REACHED) {
    if( no_messages(mb) != 0 )
        terminate(); /* ERROR */

    nTest3 = 1;
    if (nTest1*nTest2*nTest3) {
        /* Blinka lilla lysdiod */
        /* Test ok! */
    }
    terminate(); /* PASS, no receiver */
}
else
{
    terminate(); /* ERROR */
}
}

```

```

void task2(void)
{
volatile int nData_t2 = 0;
if(receive_wait(mb,&nData_t2) ==
        DEADLINE_REACHED) /* t1 kör nu */
        terminate(); /* ERROR */
if( no_messages(mb) != 0 )
        terminate(); /* ERROR */

if (nData_t2 == TEST_PATTERN_1) nTest1 = 1;
wait(20); /* t1 kör nu */

/* start test 2 */
if( no_messages(mb) != 1 )
        terminate(); /* ERROR */

if(receive_wait(mb,&nData_t2) ==
        DEADLINE_REACHED) /* t1 kör nu */
        terminate(); /* ERROR */

if( no_messages(mb) != 0 )
        terminate(); /* ERROR */

if (nData_t2 == TEST_PATTERN_2) nTest2 = 1;

/* Start test 3 */
terminate();
}

```

```

/* Task administration */
int    init_kernel( void );
int    create_task( void (* body)(), uint d );
void   terminate( void );
void   run( void );

/* Communication */
mailbox*create_mailbox( uint nMessages,
                        uint nDataSize );
action remove_mailbox( mailbox* mBox );
int     no_messages( mailbox* mBox );
exception    send_wait( mailbox* mBox,
                        void* pData );
exception    receive_wait( mailbox* mBox,
                           void* pData );
int     send_no_wait( mailbox* mBox,
                      void* pData );
int     receive_no_wait( mailbox* mBox,
                         void* pData );

/* Timing */
void    wait( uint nTicks );
void    set_ticks( uint no_of_ticks );
uint    ticks( void );
uint    deadline( void );
void    set_deadline( uint nNew );

```