* 1. Vprašanja o FreeRTOS [2T]

1. Kaj je opravilo? S katero funkcijo se ustvari? Kaj se zgodi z opravilom, ko se enkrat ustvari? Kako izgleda TCB?

Opravilo je neodvisen majhen program napisan v C kodi z neskončno zanko in brez return stavka.

Ustavari se z FreeRTOS xTaskCreate() API funkcijo.

Se mu dodeli stanje. Stanja, ki jih lahko ima opravilo:

Running : ko se opravilo dejansko izvaja.

Ready: opravila, ki niso v Blocked ali Suspended stanju ampak se ne izvajajo,

zaradi izvajanja drugega opravila z višjo ali enako prioriteto. So lahko izbrane

za izvajanje.

Blocked: opravila so v tem stanju, ko čakajo na dogodek in so premaknjena v

ready, ko se dogodek zgodi.

Suspended: opravilo je bilo namerno ustavljeno in čaka na prehod v stanje

Ready.

TCB je struktura v kateri so shranjene pomembne informacije o opravilih, ki so ključne za pravilno delovanje in njihovo razvrščanje.

Kako zgleda: <https://github.com/maniacbug/FreeRTOS/blob/master/tasks.c>

1. Opišite diagram prehajanja stanj za opravila v FreeRTOS.

Prehajanje stanj:

Running -> Ready

/ s klicem vTaskSuspend() Suspended

/ z API funkcijo Blocking Blocked

Ready -> Running

/ s klicem vTaskSuspend() Suspended

Suspended -> s klicem vTaskResume() Ready

Blocked -> z dogodkom (Event) Ready

/ s klicem vTaskSuspend() Suspended

1. Na kakšen način FreeRTOS razvrsti opravila? Katere se izvedejo najprej?

Ko se opravila ustvarijo se jim dodeli prioriteto. Opravila se razvrstijo glede na to dodeljeno prioriteto, večjo je numerično število prioritete , hitreje se opravilo izbere za izvajanje.

1. Kaj je pre-empitve in kaj co-operative scheduling? Kako se nastavi v FreeRTOS?

Pre-empitve scheduling je takoj, ko pride v ready stanje opravilo z večjo prioriteto,

kot trenutno opravilo, ki se izvaja, se opravilo z večjo prioriteto premakne v stanje izvajanja in opravilo, ki se je izvajalo, premakne v ready stanje.

V Co-operative scheduling pride do spreminjanja konteksta samo. ob prehodu

opravila iz running stanja v blocked ali ob klicu taskYIELD().

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava

Opis je samodejno ustvarjen

5) Kaj je time-slicing in kje se definira/uporablja?

Time-slicing se uporablja ko imata dva ali več opravila enako prioriteto in se opravili

Izvajata v časovnih kosih kjer se v zacetku časovnega kosa začne opravilo in na koncu kosa

konča opravilo in začne drugo. Tako se več opravil izvajaja “hkrati”.

6)Poiščite/Skopirajte kodo za preklop konteksta.

void Opravilo()

{

delajNekaj();

taskYIELD();

}

Vir: <https://www.freertos.org/a00020.html#taskYIELD>

7) Kakšen je namen datoteke FreeRTOSConfig.h?

Za konfiguracijo RTOS jedra.

8) Katere sheme alokacij pomnilnik podpira FreeRTOS?

Heap\_1 , Heap\_2 omogoča brisanje in best fit algoritem, Heap\_3 uporablja malloc() in free(),

Heap\_4 best fit algoritem , manj fragmentacije.

9) Kaj je CMSIS? Kakšne je povezava z FreeRTOS?

Common Microcontroller Software Interface Standard (CMSIS)

CMSIS omogoča preproste programske vmesnike do procesorja in njegovih zunanjih naprav, poenostavlja ponovno uporabo programske opreme, skrajša krivuljo učenja za razvijalce mikrokontrolerjev in skrajša čas za trženje novih naprav.

CMSIS zagotavlja standardizirane vmesnike in pripomočke za mikrokontrolerje, ki jih je mogoče uporabiti za izboljšanje in poenostavitev uporabe RTOS, kot je FreeRTOS. CMSIS-RTOS API je generični vmesnik RTOS za naprave s procesorjem Cortex-M. CMSIS-RTOS ponuja API za komponente programske opreme, ki zahtevajo storitve RTOS. Ta abstrakcija omogoča pisanje kode, ki je prenosljiva.