UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria de Manresa

Final de PBN

20 de juny de 2022 | Enginyeria de Sistemes TIC | 135 MINUTS

COGNOMS:

NOM: SUBGRUP:

Exercici 1 [1 punt]. Considereu una funció amb el prototip següent: `int f(float *const a)`. És correcta la sentència `a = (float *const)12` en el cos de la funció? Justifica la resposta.

Exercici 2 [1 punt]. En el llenguatge C les taules són de longitud fixa. Per contra, a Python les llistes —que és el que més recorda una taula— són de longitud dinàmica. Quines avantatges aporta el model estàtic de C?

Exercici 3 [1 punt]. Considereu un petit programa com aquest executant-se en un Arduino:

```
for(;;) PINB |= _BV(PINB0);
```

Observeu que crea un senyal quadrat al pin 0 del port B. Si compilem amb optimització, de quin ordre serà el període del senyal? Encercla el més correcte.

100 ms | 10 ms | 1 ms | 100 µs | 10 µs | 1 µs | 100 ns | 10 ns | 1 ns

Exercici 4 [1 punt]. En el context del projecte del control de semàfors, analitzeu quines parts tant del màster com de l'slave s'executen amb les interrupcions inhibides.

Exercici 5 [1 punt]. Suposeu una aplicació que usa intensivament el mòdul `timer`. Imagineu que hi ha tants *callbacks* a executar que el temps d'execució supera el temps de `TICK`. Raoneu què succeiria en un cas així.

Exercici 6 [3 punts]. El modul `timer` del projecte té aspectes millorables. Observeu aquesta situació patològica: (1) Es planifica una acció A per a ser executada al cap de 20 s i torna l'identificador i_A; (2) es cancel·la l'acció d'identificador i_A; (3) Es planifica una acció C per ser executada al cap de 20 s i casualment torna l'identificador i_A, que estava lliure. Es cancel·la de nou sobre l'identificador i_A, pensant que cancel·lem A però en realitat es cancel·la C!! Es un efecte indesitjat conseqüència del reaprofitament dels identificadors de les accions planificades. CONSELL: Feu-vos un diagrama de temps si no ho enteneu.

L'objectiu d'aquest exercici és implementar un petit `timer` simplificat però que resolgui el problema anterior. Cal que els identificadors siguin (quasi) únics.

Per simplificar l'exercici, assumiu que:

a) Només hi haurà accions planificades simples, i.e.: les que s'executen una sola vegada al cap de n ticks.

b) Hi ha una funció amb signatura **void** interrupció(**void**) que és cridada per la rutina d'interrupció cada tick. A l'exercici no cal, doncs, interactuar amb el maquinari sinó implementar aquesta funció.

c) No aturarem mai les interrupcions del timer, encara que no hi hagi accions a fer.

Aquest timer tindrà només dues operacions públiques:

```
unsigned int set_accio(int nticks, callback_t f);
void cancela(unsigned int id_accio);
```

La seva especificació és similar a la de la pràctica. set_accio() activa una acció planificada que al cap de nticks cridarà el callback f. Retorna un identificador id de l'acció. cancela() cancel·la l'acció que té identificador id_accio.

Per tal d'assignar identificadors (quasi) únics seguirem el següent algoritme:

1) El primer identificador d'acció serà el 0. Si a l'acció A_i li hem donat l'identificador i, a la següent acció que planifiquem A_{i+1} li donarem l'identificador $i + 1$.

2) Una vegada arribem a l'identificador UINT_MAX haurem esgotat tot el rang dels enters sense signe. Llavors, començarem de nou amb l'identificador 0 i anirem atorgant identificadors seqüencialment però *comprovant que l'identificador atorgat no està en ús.*

ES DEMANA:

a) Que dissenyeu l'estructura de dades del timer d'acord amb el que s'ha dit abans.

b) Que implementeu la funció **void** interrupcio(**void**).

c) Que implementeu les dues operacions del mòdul.

Només cal que escriviu la part de la implementació del mòdul i no pas el header.

Exercici 7 [2 punts]. En el context del `timer` assumiu que l'instant en que es produeix l'interrupció del timer per servir el tick i, és t_i. Assumiu també que l'acció temporitzada A s'ha d'executar en aquest tick. Per la naturalesa del mòdul `timer`, l'instant en que s'executarà l'acció A no serà t_i exactament sinó $t_i + \beta$, sent $\beta > 0$ un terme d'error que canvia segons les circumstàncies. L'objectiu d'aquest exercici és raonar sobre β.

a) β no és un valor constant: generalment canvia per a cada acció i per cada instant de temps. Quins factors influeixen en el valor de β?

b) Quines característiques de β fan que el comportament del mòdul `timer` s'acosti a un timer ideal? Com influeixen els factors anteriors en aquestes característiques?

c) Hi ha cap cas particular en que β sigui constant o s'acosti a ser constant?

d) Quines característiques haurien de tenir els callbacks per a que β fós millor?

e) Concloent amb el que has deduït sobre β, suggereix com es podria millorar el mòdul `timer` en aquest àmbit.