

Control de Laboratori

Grup 20

23 de Març

Contesteu al mateix full

Nom i Cognoms:

1. (5 punts) Seleccioneu la resposta (una només) que considereu correcta en cadascun dels apartats. Cada resposta correcta val 0,5 punts. Cada resposta incorrecta resta 0,25.
 - (a) L'opció *-L* serveix per:
 - Indicar al compilador un directori on trobar llibreries
 - Indicar al Makefile un directori que conté llibreries
 - Indicar al compilador i al Makefile un directori que conté llibreries
 - Cap de les anteriors
 - (b) La rutina *escriure_pantalla_sis* retorna:
 - Un enter
 - Res
 - Un byte
 - Cap de les anteriors.
 - (c) La rutina *salvar* retorna:
 - Un punter al *top* de la pila
 - Un punter al *top* del context d'execució de l'usuari
 - No retorna res
 - El valor del registre SP
 - (d) Per a cridar una funció en C des d'un codi en ensamblador cal:
 - Declarar la funció com a *extern* en el codi C
 - Donar a la funció un nom que comenci per *_* (ex: *_nom* per a la funció *nom*)
 - Declarar la funció com a *extrn* en el codi ensamblador
 - Totes les anteriors
 - (e) Per a generar un executable a partir d'un fitxer C que crida una rutina implementada en un fitxer en ensamblador hem de:
 - Compilar el fitxer ensamblador i posar el fitxer resultant com un objecte més a la línia de *linkatge*.
 - Afegir un include del fitxer ensamblador en el fitxer C que el crida
 - Fer un fitxer de capçalera per al fitxer en ensamblador, incloure'l en el fitxer en C i deixar que el compilador trobi el fitxer ensamblador a través de les capçaleres
 - Fer servir la opció *-I* del compilador per a indicar-li on estan els fitxers auxiliars necessaris per a produir un executable

- (f) Si en un makefile volem expressar una dependència amb un fitxer f que es troba en un directori X , haurem de:
- Indicar al makefile que la dependència és amb el fitxer X/f
 - Usar la opció $-IX$ o $-LX$ del compil·lador per a indicar on trobar el fitxer, i indicar la dependència amb el fitxer f sense fer referència al directori X .
 - Usar la opció $-IX$ i $-LX$ del compil·lador per a indicar on trobar el fitxer, i indicar la dependència amb el fitxer f sense fer referència al directori X .
 - Usar la opció $-IX$ de l'eina *make* per a indicar on trobar el fitxer, i indicar la dependència amb el fitxer f sense fer referència al directori X .
- (g) L'estructura brossa es fa servir per:
- Tenir una manera d'accedir als paràmetres passats des de la ONIONLIB
 - Enganyar al compil·lador i fer-li creure que no hi ha paràmetres en la crida
 - Tenir una manera d'accedir al registre AX del context d'execució salvat en la rutina *salvar*
 - Cap de les anteriors
- (h) La funció *escriure_pantalla_bfs* rep com a paràmetres:
- Cap, perquè no fa res més que cridar a *escriure_pantalla_nvuc*
 - Cap, perquè ja estan a la pila i els pot accedir gràcies al paràmetre brossa
 - Els mateixos que la rutina *escriure_pantalla* de la ONIONLIB
 - Els mateixos que la rutina *escriure_pantalla* de la ONIONLIB més un primer paràmetre *brossa* per a poder accedir als paràmetres que ja hi ha a la pila
- (i) En cas d'arribar un valor de servei no conegut a la funció *trap*:
- El Sistema retorna un error a l'usuari
 - El Sistema ignora la petició ja que no sabem què vol l'usuari
 - El Sistema escriu un missatge per pantalla indicant l'error
 - No pot arribar un valor incorrecte ja que només hi ha una crida a llibreria de Sistema i té un valor correcte
- (j) Quan la rutina *restaurar* fa un *iret* com a última instrucció del seu codi és per a:
- Retornar al punt del codi d'usuari (fluxes) on s'ha fet una crida a sistema
 - *restaurar* no es una RAI i per tant no fa un *iret* sinó un *ret*
 - Retornar al punt de la llibreria (onionlib) on s'ha fet el *call* a una crida a sistema
 - Retornar al punt de la llibreria (onionlib) on s'ha fet el *trap*
2. (2 punts) Suposem que necessitem una estructura brossa de 28 bytes per a poder accedir als paràmetres de la crida *escriure_pantalla* des de la rutina *escriure_pantalla_sis*. Si ara definim 4 enters com a variables locals en la rutina *trap*, es veurà afectada la mida de l'estructura brossa? Per què? Raona la resposta.

3. (2 punts) Se'ns dóna una pràctica d'Onion que en executar la crida *escriure_pantalla* amb uns paràmetres correctes, sempre retorna error (-1).
- (a) Quin error creus que hi ha en aquesta pràctica si les comprovacions fetes a *escriure_pantalla_nuc* són correctes? Per què?
- (b) Què podries fer per descobrir si tens raó fent servir el debugger?
4. (1 punts) Si volguéssim empilar paràmetres a la pila en les crides de *escriure_pantalla_sis* a *escriure_pantalla_bfs* i de *escriure_pantalla_bfs* a *escriure_pantalla_nuc* però seguir tenint accés als paràmetres passats per l'usuari a la crida a sistema sense tornar-los a ficar a la pila, com ho podríem fer? Què ens caldria?