

Control de Laboratori (grup 40)

25 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 que conté llibreries
 - Indicar a l'eina *make* un directori que conté llibreries
 - Indicar al compilador una llibreria amb la qual ha de linkar
 - Indicar a l'eina *make* una llibreria amb la qual ha de linkar
 - (b) La crida a la rutina *escriure_pantalla_sis* s'ha de fer:
 - Recollint abans els paràmetres de la pila i passant-los com a arguments de la rutina *escriure_pantalla_sis*
 - Sense paràmetres, perquè aquests ja estan a la pila, situats correctament perquè la rutina *escriure_pantalla_sis* pugui recollir-los sense necessitat de paràmetre *brossa*
 - Sense paràmetres, perquè aquests ja estan a la pila, i seran recollits per la rutina de la capa Nucli
 - Cap de les anteriors.
 - (c) 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
 - La rutina *restaurar* no és una RSI 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*
 - (d) Volem que la llibreria *onionlib.lib* s'actualitzi cada cop que es modifiqui el fitxer *onionlib.asm*. Què s'ha de fer en el Makefile per garantir aquest comportament?
 - Afegir una dependència al Makefile del fitxer *onionlib.obj* respecte el fitxer *onionlib.asm* i compilar el fitxer *onionlib.asm*
 - Afegir una dependència al Makefile del fitxer *onionlib.lib* respecte al fitxer *onionlib.obj*
 - Reemplaçar l'objecte de la llibreria amb la comanda *tlib lib\onionlib.lib +- onionlib.obj*
 - Totes les anteriors
 - (e) Disposem d'una rutina en assembler declarada com a *near*, que accedeix a un paràmetre amb la instrucció *mov ax, word ptr [bp+4]*. Com hauria d'accedir a aquest paràmetre si estigués declarada com a *far* enlloc de *near*?
 - Hauria d'accedir de la mateixa manera
 - L'accés seria amb *mov ax, word ptr [bp+6]*, donat que pels procediments *far*, l'adreça de retorn ocupa 4 bytes
 - L'accés seria amb *mov ax, word ptr [bp+6]*, donat que pels procediments *far*, el registre IP ocupa 4 bytes
 - Cap de les anteriors

- (f) Ens apareix el següent missatge **Error: Undefined symbol _restaurar in module sistema.c** quan fem *make*. Què ens està passant?
- No hem fet l'include del fitxer de capçalera on està definida la rutina *restaurar* al fitxer *sistema.c*
 - No estem linkant amb l'objecte on està definida la rutina *restaurar*
 - No hem fet la declaració *extern void restaurar()* al fitxer *sistema.c*
 - Cap de les anteriors
- (g) Com es retorna el resultat de la crida a *sistema* a la rutina *escriure_pantalla_nuc*?
- Es fa un *return*
 - S'actualitza el contingut de la posició del context del procés ocupada pel registre AX utilitzant l'estructura *brossa*
 - S'actualitza el contingut de la posició del context del procés ocupada pel registre AX utilitzant un apuntador al context del procés
 - Cap de les anteriors
- (h) Donat que l'únic que fa la rutina *escriure_pantalla_bfs* és cridar a *escriure_pantalla_nuc*, podem cridar directament a *escriure_pantalla_nuc* des de *escriure_pantalla_sis*?
- Sí, de fet s'hauria de fer d'aquesta manera per millorar el rendiment del sistema operatiu
 - Sí, perquè tots els símbols de la capa Nucli són accessibles des de les capes superiors
 - No, perquè tots els símbols de la capa Nucli només són accessibles des de la capa BFS
 - No, perquè els símbols públics de la capa Nucli només són accessibles des de la capa BFS
- (i) Al entrar al trap, respecte a les interrupcions, hem de:
- No cal inhibir les interrupcions ja que aquestes no es poden produir dins d'una RSI
 - No cal inhibir les interrupcions perquè s'inhibeixen automàticament quan es fa el trap
 - Inhibir les interrupcions perquè no ens interrompin mentre estem dintre del Sistema
 - Cap de les anteriors
- (j) 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
2. (2 punts) Dibuixa el contingut de la pila just abans de cridar la rutina *escriure_pantalla_sis*. Suposa que la rutina *salvar* empila els registres DX, CX, BX, AX, SI, DI, ES, DS, BP (en aquest ordre). Com accediries al contingut del registre AX? (suposa que tenim una variable *int *ctx* que apunta al context guardat per la rutina *salvar*).

3. (2 punts) Se'ns dóna una pràctica d'Onion que en executar la crida *escriure_pantalla* amb uns paràmetres correctes retorna valors aleatoris.
- (a) Quin error creus que hi ha en aquesta pràctica si la comprovació de paràmetres feta a *escriure_pantalla_nuc* és correcta? Per què?
- (b) Què podries fer per descobrir si tens raó fent servir el debugger?
4. (1 punts) Volem canviar el mecanisme de pas de paràmetres a la pràctica de manera que tinguem un registre que serveixi d'apuntador als paràmetres. Explica breument quines funcions caldria modificar i de quina manera.