PARCIALITOS TIPO DE DATOS AVANZADOS, HEAP, GITHUB Y RECURSIVIDAD

Pregunta 1 2 de 2 puntos

La herramienta git tiene ventajas respecto a los sistemas de almacenamiento en la nube (como Dropbox, Google Drive, OneDrive, etc.). Indique una de estas ventajas, explicándola.

Pregunta 2 5 de 5 puntos

Declarar el alias llamado `data_t` a una estructura que tenga 2 campos:

- `pfun`: puntero a función que recibe dos punteros genéricos y devuelve un int.
- `arr`: arreglo de 3 int.

Luego definir una variable de la estructura anterior y llamarla `var`.

Finalmente asignarle su número de legajo al último elemento del arreglo de la variable anterior.

Pregunta 3 3 de 3 puntos

Indicar qué devuelve la siguiente función recursiva si se la invoca con **8**. Justificar la solución con el cálculo de la respuesta.

```
unsigned int rec (unsigned int x)
{
    if (x <= 3)
        return x;
    else
        return (recursiva(x-2) + recursiva(x-3));
}
```

viernes 21 de mayo de 2021 16H36' ART

Pregunta 1 6 de 6 puntos

Indicar todos los problemas del siguiente programa respecto al uso de memoria dinámica, considerando las buenas prácticas de programación. Luego reescribir el programa corregido.

```
#include <stdio.h>
int main (void)
{
    int i, *p;
    p = malloc(9);
    for (i=0 ; i<=9 ; i++)
        p[i] = -1;
    return 0;
}
```

Respuesta seleccionada:

```
#include <stdio.h>
//Le falta incluir la libreria stdlib
int main (void)
{
    int i, *p;
    p = malloc(9);//en vez de reservar 9 bytes seria mejor reservar 9 sizeof(int), y castear explicitamente malloc ya que es un void*
    for (i=0 ; i<=9 ; i++)//no verifica que se haya podido asignar la memoria
        p[i] = -1;
    return 0;
//Hay que liberar la memoria!!
}

#include <stdio.h>
#include <stdlib.h>
int main (void)
{
    int i, *p;
    p = (int *) malloc(9*sizeof(int));
    if (p==NULL)
    {
       printf("Error.Memory not allocated.\n");
    }
    else {
    for (i=0 ; i<=9 ; i++){
        p[i] = -1;
     }
    }
    free(p); //borro la memoria igual por buena practica, aunque no tiene mucho sentido honestamente si no llegue a trabajar con los numeros guardados. Deberian haberse usado o print
    return 0;
}
```

Comentarios para respuesta: No es necesario realizar casteo explicito a int*, el puntero generico se castea implícitamente al asignarlo sin problemas.

Pregunta 2 2 de 2 puntos

Explicar al menos 2 diferencias entre las funciones `fopen()` y `open()` para el manejo de GPIOs.

Respuesta seleccionada: La funcion fopen () es portable mientras que open () no. Esto quiere decir que la portable esta disponible en la mayoria de las plataformas y siempre con el mismo prototipo mientras que open (), no.
Luego, la funcion fopen () resulta mas rapida por el uso del buffer, mientras que open () no usa buffer y resulta siendo mas lenta. El buffer utiliza memoria RAM para almacenar informacion temporaria, y es rapido y eficiente. Por otro lado, que sea unbuffered open () hace que no sea eficiente ya que el acceso al disco rigido toma un tiempo mayor que acceder a la RAM. En el caso del manejo de GPIOs la diferencia no es notoria ya que simplemente escribimos un caracter por vez para cambiar el estado del pin.
fopen() es de alto nivel y open () de bajo nivel porque open es un wrapper (version simplificada de una funcion) de un system call del kernel, y por ende esta mas cercano al hardware, mientras que fopen () tiene un mayor nivel de abstraccion.

Comentarios para respuesta: [No se ha dado ninguna]

Pregunta 3 1 de 2 puntos

Explicar que es el *argument promotion*. Dar un ejemplo concreto de uso.

Respuesta seleccionada: Cuando se invoca a una fuuncion de argumentos variables, estos argumentos experimentan promociones automaticas a otros tipos de datos. char y short se promueven a int, y float a double. Esto es porque ninguna funcion puede recibir argumentos de tipo char, float o short.

Comentarios para respuesta: Falta un ejemplo concreto

viernes 21 de mayo de 2021 16H37' ART

Pregunta 1

0 de 3 puntos

Crear un alias al tipo de datos puntero a un arreglo de 10 enteros.

Respuesta seleccionada: typedef int (int*) [10] MYPUN:

Comentarios para respuesta: Alias mal creado (nombre mal colocado)

arreglo de punteros (en lugar de puntero de arreglos).

Repite int (2 veces).

Pregunta 2

5 de 5 puntos

Dado el siguiente programa que es compilado en una arquitectura big-endian de 16 bits:

a. ¿Compila el siguiente programa? De no ser así, indicar porqué y corregirlo.
b. Indicar cuantos bytes ocupa la variable `var`.
c. Indicar qué imprime el siguiente programa.

```
#include <stdio.h>

union {
  unsigned int u;
  int    i;
  char   c;
} var;

int main(void)
{
  var.i = 0xFACE;
  var.c = 0x37;
  printf("%4X", var.i);
  return 0;
}
```

Respuesta seleccionada:
```
1) Si, compila
2) Ocupa 2 bytes
3) Imprime 37CE
```

Comentarios para respuesta: OK