

## CC3301 Programación de Software de Sistemas

### Tarea 2 – Semestre Primavera 2018 – Prof.: Luis Mateu

En esta tarea Ud. deberá implementar una cola de prioridades genérica usando *listas enlazadas*. Para ello Ud. deberá programar la función *nuevaColaPriLista* en el archivo **colapri-lista.c**:

```
typedef int (*CompPri)(void *p, void *q);  
ColaPri nuevaColaPriLista(CompPri comp);
```

Ud. no sabe nada acerca del tipo de los elementos almacenados en la cola. Solo dispone de un puntero *comp* a una función que compara 2 elementos, la que entrega  $< 0$ ,  $0$  o  $> 0$  si es que el primer elemento tiene mejor, igual o peor prioridad que el segundo.

### Uso de la cola de prioridades

Supongamos que los elementos son *strings* y la prioridad está dada por el orden alfabético y por lo tanto la función *strcmp* sirve para comparar sus prioridades. Entonces la cola de prioridades se crea mediante:

```
ColaPri c= nuevaColaPriLista((CompPri)strcmp);
```

Esta tarea va asociada con un conjunto de archivos que Ud. debe descargar de U-cursos. La definición de la estructura de una cola de prioridades genérica está en **colapri.h**. El único campo visible para quienes usan este tipo de datos abstracto es *ops*, que consiste en un puntero a una estructura con punteros a las funciones que manipulan la cola. El encabezado de *nuevaColaPriLista* se encuentra en **colapri-lista.h**.

La cola de prioridades debe implementar las operaciones *agregar*, *extraer*, *mejor*, *tamano*, *iterador* y *destruir*. El iterador es de tipo *IterColaP* e implementa las operaciones *continua*, *proximo* y *destruir*. Por ejemplo:

```
c->ops->agregar(c, "perro"); /* agrega un elemento a la cola */  
c->ops->agregar(c, "gato");  
char *best= c->ops->mejor(c); /* elemento con mejor prioridad: gato */  
int n= c->ops->tamano(c); /* cantidad de elementos en la cola: 2 */  
c->ops->agregar(c, "pez");  
IterColaP iter= c->ops->iterador(c);  
while (iter->ops->continua(iter)) /* muestra: gato perro pez */  
    printf("%s\n", (char*)iter->ops->proximo(iter));  
iter->ops->destruir(iter);  
char *e1= c->ops->extraer(c); /* gato */  
char *e2= c->ops->extraer(c); /* perro */  
char *e3= c->ops->extraer(c); /* pez */  
c->ops->destruir(d); /* destruye la cola de prioridades */
```

Observe que el iterador entrega todos los elementos de la cola *por orden de prioridad* pero sin extraerlos. La operación *extraer* entrega y elimina el elemento con mejor prioridad.

Implemente todas estas operaciones en el archivo **colapri-lista.c**. Entenderá mejor como resolver esta tarea estudiando el archivo **colapri-arreglo.c** que contiene una implementación trivial de la cola basada en arreglos.

Pruebe su implementación basada en listas enlazadas con el archivo **test-colapri-lista.c**. Este test verifica que su implementación entrega los mismos resultados que la que se encuentra en **colapri-arreglo.c**.

### Recursos

Descargue de U-cursos el archivo **t2.zip** que contiene los archivos necesarios para resolver esta tarea. Lea el archivo README.txt con más instrucciones para esta tarea.

La cola de prioridades genérica de esta tarea es similar a los diccionarios genéricos que se estudiaron en la clase auxiliar del Viernes 26 de octubre. Estudie esa clase auxiliar.

### Entrega

Entregue su tarea solo si funciona correctamente con **test-colapri-lista.c** en la máquina *anakena.dcc.uchile.cl* y compila sin arrojar warnings. Ud. debe entregar el archivo **colapri-lista.c** por medio de U-cursos. Se descontará medio punto por día de atraso. No se consideran los días sábado, domingo, festivos o vacaciones.