CC4302 – Sistemas Operativos
Pauta Auxiliar 3

Profesor: Luis Mateu
Auxiliar: Diego Madariaga

8 de abril de 2020

1. Pregunta 1

```
#define P 4
typedef struct {
  Pizza *pizza;
  int cocida;
} Orden;

nMonitor ctrl;
nTask pizzeriaTask;
FifoQueue *pendientes;

void iniciarPizzeria() {
  ctrl= nMakeMonitor();
  pizzeriaTask= nEmitTask(hornoProc);
  pendientes= MakeFifoQueue();
}

Orden* ordenarPizza() {
  Orden *po= (Orden*)nMalloc(sizeof(Orden));
  po->pizza= obtenerPizzaCruda();
  po->cocida= FALSE;
  nEnter(ctrl);
  PutObj(pendientes, po);
  nNotifyAll(ctrl);
  nExit(ctrl);
  return po;
}

Pizza* esperarPizza(Orden *po) {
  nEnter(ctrl);
  while (!po->cocida)
    nWait(ctrl);
  nExit(ctrl);
  return po->pizza;
}

Pizza* comprarPizza() {
  Orden *o = ordenarPizza();
```

```
  Pizza *p = esperarPizza(o)
  return p;
}

int hornoProc() {
  for(;;) {
    int k= 0, i;
    Pizza *pizza_vec[P];
    Orden *orden_vec[P];
    nEnter(ctrl);
    while (EmptyFifoQueue(pendientes))
      nWait(ctrl);
    while (!EmptyFifoQueue(pendientes) && k<P)
      orden_vec[k]= (Orden*)GetObj(pendientes);
      pizza_vec[k]= orden_vec[k]->pizza;
      k++;
    }
    nExit(ctrl);

    hornear(horno, pizza_vec, k);

    nEnter(ctrl);
    for (i=0; i<P; i++)
      orden_vec[i]->cocida= TRUE;
    nNotifyAll(ctrl);
    nExit(ctrl);
  }
  return 0;
}
```

2. Pregunta 2

```
Camion *camiones [P];    // la flota de camiones
int ocupados [P];        // ocupados[k] es verdadero si camiones[k] esta siendo
                         // ocupado
Ciudad *ubic [P];        // si el k-esimo camion esta ocioso, indica en que
                         // ciudad quedo
nMonitor m ;             // = nMakeMonitor() ;

void transportar(Contenedor *cont, Ciudad *orig, Ciudad *dest) {
        int mejor = -1, k ;
        nEnter(m);
        for (;;) {       // buscar el mas cercano
                for (k = 0; k < P; k ++)
                        if (!ocupados[k] && (mejor <0 || distancia(ubic[k], orig) <
                            distancia(ubic[mejor], orig)))
                                mejor = k;
                if (mejor >=0)
                        break;
                nWait(m);
        }
        ocupados[mejor] = TRUE ;
        nExit(m) ;
        // conducir, cargar, conducir y descargar fuera de la seccion critica
        conducir(camiones[mejor], ubic[mejor], orig);
        cargar(camiones[mejor], cont);
        conducir(camiones[mejor], orig, dest);
```

```
        descargar(camiones[mejor], cont);
        nEnter(m);
        ocupados[mejor] = FALSE;
        ubic[mejor] = dest;
        nNotifyAll(m);
        nExit(m);
}
```