Pregunta 1

Parte a.- Programe la función *integral* que calcula numéricamente la integral de una función que se recibe como parámetro. El encabezado de la función es:

```
typedef double (*Funcion)(void *p, double x);
double integral(Funcion f, void *p,
                double xi, double xf, int n);
```

Ud. debe calcular numéricamente $\int_{xi}^{xf} f(p,x)\,dx$ usando el método de los trapecios, en donde *n* es el número de trapecios usados para aproximar el área bajo la curva. Para ello use la siguiente fórmula:

$$\int_{xi}^{xf} f(p,x)\,dx \approx h\cdot[\frac{f(p,xi)+f(p,xf)}{2}+\sum_{k=1}^{n-1} f(p,xi+k\cdot h)]$$

con $h=\frac{xf-xi}{n}$. El puntero p se usa para pasar parámetros adicionales a la función en caso de necesidad. A modo de ejemplo suponga que Ud. dispone de la función *g(x, y)*. En el código de más abajo la función *integral_g_dx* usa *integral* para calcular numéricamente $\int_{xi}^{xf} g(x,y)\,dx$. Observe que como *g* no posee el tipo requerido por *integral*, se introduce *g_aux* que sí posee el tipo requerido.

```
double g_aux(void *ptr, double x) {
  double y= *(double *)ptr;
  return g(x, y);
}

double integral_g_dx(double xi, double xf, double y, int n) {
  return integral(g_aux, &y, xi, xf, n);
}
```

Parte b.- Programe la función *integral_g_dx_dy* que calcula numéricamente $\int_{yi}^{yf}\int_{xi}^{xf} g(x,y)\,dx\,dy$. Para ello use las funciones *integral* e *integral_g_dx* de la parte a. El encabezado de la función es:

```
double integral_g_dx_dy(double xi, double xf, int n,
                        double yi, double yf, int m);
```

en donde *n* es el número de trapecios a usar para la variable *x* y *m* es el número de trapecios para la variable *y*.

Parte c.- Programe la función *integral_g_dx_dy_par* que calcula la misma integral de la parte b pero en paralelo usando 8 cores. *Hint*: subdivida el intervalo [yi, yf] en 8 subintervalos y use 8 threads para calcular la integral de cada subintervalo por medio de la función *integral_g_dx_dy* de la parte b. Suponga que *m* es múltiplo de 8.

Pregunta 2

Parte i.- La función *imprimir_asinc* le permite a *múltiples threads* solicitar la impresión de documentos sin tener que esperar que terminen de imprimirse. Los documentos solo se encolan para que 2 threads de servicio, que ejecutan la función *hilo_impresor*, los impriman más tarde en alguna de las 2 impresoras disponibles por orden de llegada. Para ello invocan la función *imprimir* que recibe el documento y el número de la impresora a utilizar (0 o 1). La función *imprimir* toma un tiempo considerable. La función *imprimir_asinc* retorna un recibo que se usará más tarde para confirmar que el documento fue finalmente impreso. Esto se hace pasando el recibo a la función *confirmar*, la que retorna de inmediato si el documento terminó de imprimirse o espera hasta su finalización. La siguiente es una implementación incorrecta e ineficiente de estas funciones:

```
typedef struct {
  Doc *doc;
  int listo;
} Rec; /* Un recibo */
ColaFifo *cola;
Rec *imprimir_asinc(Doc *doc) {
  Rec *prec=
        malloc(sizeof(Rec));
  prec->doc= doc;
  prec->listo= FALSE;
  agregar(cola, prec);
  return prec;
}
```

```
void confirmar(Rec *prec) {
  while (!prec->listo)
    ;
  free(prec);
}
void hilo_impresor(int id) {
  for (;;) {
    while (vacia(cola))
      ;
    Rec *prec= extraer(cola);
    imprimir(prec->doc, id);
    prec->listo= TRUE;
} }
```

Reprograme el código de más arriba de manera correcta y eficiente. Ud. no puede hacer *busy-waiting*. La función *imprimir* y la cola fifo son dadas. *Hint*: tome esta implementación como referencia de la funcionalidad pedida. Está casi buena pero le falta la sincronización. Requiere pocas modificaciones.

Parte ii.- Se necesita agregar la función *imprimir_urgente* que imprime un documento en forma prioritaria. Esto significa que solo se puede iniciar la impresión de un documento no urgente cuando no hay documentos urgentes pendientes. La función *imprimir_urgente* solo retorna una vez que el documento fue impreso. Su encabezado es:

```
void imprimir_urgente(Doc *doc);
```

Programe esta nueva funcionalidad. Ud. necesitará reprogramar algunas de las funciones de la parte i. *Hint*: use una segunda cola fifo para los documentos urgentes.