



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Departamento de electrónica

Informática I

Resumen de funciones V.1
WWW.UTNIANOS.COM.AR
F.E.P

Realizado por: Fernando (F.E.P)
www.UTNianos.com.ar

(Actualizada al 3-02-13)



FUNCIONES DE LIBRERIAS INFORMÁTICA I - ELECTRONICA

LENGUAJE C

Librería stdio.h:

- Funciones de entrada/salida

int getchar(void): Lee el siguiente carácter de la entrada estándar y lo devuelve como entero.

char *gets(char *s): Lee el siguiente carácter de entrada y lo coloca en el arreglo s hasta que encuentra un carácter de nueva línea. Agrega carácter de terminación nulo al arreglo.

int putchar(int c): Imprime el caracter almacenado en c.

int puts(const char *s): Imprime la cadena s seguida por el carácter de nueva línea.

- Funciones de archivos

FILE *fopen(const char *nombre, const char *modo): Abre un archivo cuyo nombre es la cadena apuntada por nombre y adjudica un flujo a ello, modo indica modo de operación del archivo (ver agregado al final). Retorna NULL en caso de error.

int fclose(FILE *Fp): Cierra el archive asignado en el FILE*. Retorna 0- éxito o NULL- falla.

int feof(FILE *Fp): Verifica el final del archivo. Fin del archivo carácter distinto de 0, posición dentro del archivo 0. Ejemplo:
while(!feof(fp))

void rewind(FILE *Fp): Ubica el cursor al inicio del archivo
`int rename(const char *viejo, const char *nuevo);`

char *fgets(char *cadena, int n, FILE *Fp): Copia una cadena de caracteres desde el archivo al buffer(cadena). N- Cantidad máxima a leer -1. Se detiene cuando lee n-1 caracteres, finaliza el archivo, lee un '\n' y agrega al final de la cadena el '\0'



int fgetc(FILE *Fp): Lee un character desde un dispositivo. Retorna el carácter leído.

int fscanf(FILE *Fp, const char *formato, variables...): Lectura formateada desde un flujo. Retorna: Cantidad de campos leídos, convertidos y guardados (éxito)

int fputs(const char *cadena, FILE *Fd): Copia una cadena de caracteres desde el buffer (cadena) al archivo. Retorna ultimo carácter leído (éxito).

int fputc(int c, FILE *Fp): Escribe un carácter en un dispositivo. Retorna carácter escrito.

int fprintf(FILE *Fp, const char *formato, variables...): Escritura formateada desde un flujo. Retorna cantidad de bytes escritos (éxito)

size_t fread(void *Buffer, size_t tamaño, size_t cantidad, FILE *Fp): Lectura de un bloque de datos binarios desde un flujo. Retorna cantidad de bloques leídos (éxito)

size_t fwrite(const void *puntero, size_t cantidad, size_t cantidad, FILE *Fp): Escritura de un bloque de datos binarios desde el flujo. Retorna cantidad de bloques escritos (éxito)

Librería stdlib.h:

double atof(const char *PtrN): Convierte la cadena ptrN a double.

int atoi(const char *PtrN): Convierte la cadena ptrN a int.

long int atol(const char *PtrN): Convierte la cadena ptrN a long.

void free(void *ptr): Libera el espacio de memoria apuntado por ptr, que debe haber sido obtenida por una llamada previa de malloc, calloc o realloc.

void *malloc(size_t tamaño): Asigna size_t tamaño y devuelve un puntero a la memoria asignada.

int rand(void): La function rand calcula una secuencia de números enteros pseudo-aleatorios en el intervalo 0 a RAND_MAX. Rand()%n calcula un valor entre 0 y n. Retorna dicho valor.



void srand(unsigned int semilla): Semilla es un valor para una secuencia nueva de números, si la misma es fija rand() genera siempre la misma secuencia pseudo-aleatoria. Ejemplo:
srand(time(NULL)).

void *realloc(void *ptr, size_t tamaño): Cambia el tamaño del bloque de memoria apuntado por ptr a size_t tamaño. El contenido permanece inalterado si el nuevo tamaño es mayor o igual que el tamaño original.

int sizeof(var): sizeof calcula el tamaño en bytes que se encuentra en el argumento. Retorna dicho valor.

Librería string.h:

char *strcat(char*s1,const char *s2): Agrega la cadena s2 al arreglo s1. El primer carácter de s2 sobrescribe el carácter de terminación nulo de s1. Devuelve el valor de s1. El carácter nulo se agrega de la segunda cadena.

char *strncat(char*s1,const char *s2,size_t n): Agrega al menos n caracteres desde la cadena s2 al arreglo s1. Devuelve el valor de s1. El carácter nulo se agrega automáticamente.

int strcmp(const char *s1,const char *s2): Compara la cadena s1 con la cadena s2. La función devuelve 0, menor que 0, o mayor que 0, si s1 es igual, menor, o mayor que s2, respectivamente.

int strncmp(const char *s1,const char *s2,size_t n): Compara hasta n caracteres de la cadena s1 con la cadena s2. La función devuelve 0, menor que 0, o mayor que 0, si s1 es igual, menor, o mayor que s2, respectivamente.

char *strcpy(char *s1,const char *s2): Copia la cadena S2 dentro del arreglo s1. Devuelve s1.

char *strncpy(char *s1, const char *s2,size_t n): Copia al menos n caracteres de la cadena s2 dentro del arreglo s1. Devuelve el valor de s1. Se debe agregar el carácter nulo.

size_t strlen(const char *s): Determina la longitud de la cadena s. Devuelve el numero de caracteres que preceden al carácter de terminación nulo.



char *strstr(const char *s1, const char *s2): Localiza la primera ocurrencia en la cadena s1 de la cadena s2. Si se localiza la cadena, se devuelve un apuntador a la cadena en s1. De lo contrario se devuelve un apuntador a NULL.

Librería math.h:

double pow(double x, double y): Calcula x elevado a la potencia y. Devuelve x^y

double sqrt(double x): Calcula la raíz cuadrada del valor no negativo x. Devuelve \sqrt{x}

Librería ctype.h:

int isupper(int c): Devuelve un valor verdadero si c es una letra mayúscula, de lo contrario devuelve 0.

int islower(int c): Devuelve un valor verdadero si c es una letra minúscula, de lo contrario devuelve 0.

int toupper(int c): Si c es una letra minúscula devuelve c como una letra mayúscula, de lo contrario devuelve c sin modificación.

int tolower(int ch): Si c es una letra mayúscula devuelve c como una letra minúscula, de lo contrario devuelve c sin modificación.



Librería sock-lib.h

int conectar (int argc, char * argv[]): Crea el socket, lo configura, se conecta y devuelve un socket de comunicación.

int Open_conection (struct sockaddr_in * my_addr): Esta función crea el socket, enlaza el socket con la ip y puerto y pone el socket a la escucha. Devuelve el socket que recibe conexiones.

int Aceptar_pedidos (int sockfd): Acepta la conexión entrante. Sockfd es el socket que contiene las conexiones. Retorna el socket para intercambio de datos.

close(sockfd): Cierra el socket.

int read(int sockfd, void *buffer, int nbytes): La función lee nbytes del flujo de datos asociado, los coloca en buffer. Retorna caracteres leídos -éxito ó -1 -error.

Write

int write(int sockfd, void *buffer, int nbytes): La function escribe nbytes desde el buffer en el flujo de datos asociad. Retorna caracteres escritos -éxito ó -1 indicando error.