CSI 402 – Systems Programming – Handout 12.3

A Simple Example for Function Pointers

Note: The purpose of this handout is to present a simple example to show how function pointers are used in a C program. This example consists of four functions called `main`, `compute`, `linear` and `quadratic`. One of the parameters of the `compute` function is a function pointer. Note that `main` calls `compute` twice. The first call passes `linear` as the function parameter. The second call passes `quadratic`.

```
#include  <stdio.h>

#define  START_VAL   0
#define  END_VAL     3
#define  STEP_SIZE   1

/* Function prototypes. */

int  linear (int x);
int  quadratic (int x);

/* The first parameter of the following function is a */
/* pointer to a function.                            */

void  compute(int (*f)(int x), int sval, int fval, int ssize);

int main(void) {

  /* In each call to function compute, the first parameter */
  /* is the name of a function.                            */

  compute(linear, START_VAL, END_VAL, STEP_SIZE);
  printf("\n");
  compute(quadratic, START_VAL, END_VAL, STEP_SIZE);

  return 0;
} /* End of main. */
```

(over)

```
void  compute( int (*f)(int x), int sval, int fval, int ssize) {

  /* This function computes and prints the value of the function *f */
  /* for the values in the range [sval .. fval] using the step size */
  /* given by the parameter ssize.                                  */

  int i, temp;
  for (i = sval; i <= fval; i += ssize) {
     temp = (*f)(i);
     printf("%d  %d\n", i, temp);
  }
} /* End of compute. */


int  linear (int x) {

  /* Computes and returns the value of a linear function */
  /* at the point given by the parameter x.              */

  #define  LCOEFF1  6
  #define  LCOEFF2  5

  return (LCOEFF1 * x + LCOEFF2);
} /* End of linear. */


int  quadratic (int x) {

  /* Computes and returns the value of a quadratic function */
  /* at the point given by the parameter x.                 */

  #define  QCOEFF1  6
  #define  QCOEFF2  3
  #define  QCOEFF3  2

  return ((QCOEFF1 * x + QCOEFF2) * x + QCOEFF3);
} /* End of quadratic. */
```

Output:

```
0  5
1  11
2  17
3  23

0  2
1  11
2  32
3  65
```