```
 1: #!/bin/sh
 2: # $Id: mk,v 1.8 2014-04-23 21:26:13-07 - - $
 3: #
 4: # This script takes the names of C source files and compiles them
 5: # into executable images.  Each source must be a complete program.
 6: #
 7:
 8: export PATH=$PATH:/afs/cats.ucsc.edu/courses/cmps012b-wm/bin
 9: GCC='gcc -g -O0 -Wall -Wextra -std=gnu99'
10:
11: function label() {
12:    echo ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
13:    echo "$@"
14:    echo ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
15: }
16:
17: function catnv() {
18:    for arg in $*
19:    do
20:       if [ "$arg" = "-f" ]
21:       then
22:          echo -e '\f'
23:       else
24:          label $arg
25:          cat -nv $arg
26:       fi
27:    done
28: }
29:
30: for CSOURCE in *.c
31: do
32:    EXECBIN=${CSOURCE%.c}
33:    if [ "$CSOURCE" == "$EXECBIN" ]
34:    then
35:       echo $0: $CSOURCE: is not a C source file
36:    else
37:       echo CSOURCE=$CSOURCE EXECBIN=$EXECBIN
38:       set -x
39:       cid + $CSOURCE
40:       checksource $CSOURCE
41:       catnv $CSOURCE >$CSOURCE.log
42:       label "Errors: $GCC $CSOURCE -o $EXECBIN" >>$CSOURCE.log
43:       $GCC $CSOURCE -o $EXECBIN -lm >>$CSOURCE.log 2>&1
44:       cat $CSOURCE.log
45:       set +x
46:       echo ============================================================
47:    fi
48: done
49:
50: mkpspdf Listing.ps $0 *.log
51:
```

```
  1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  2: environ.c
  3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  4:      1  // Name: Kelly Scanlon
  5:      2  // User: kpscanlo
  6:      3
  7:      4
  8:      5  #include <stdio.h>
  9:      6
 10:      7  extern char **environ;
 11:      8
 12:      9  int main (int argc, char **argv) {
 13:     10     (void) argc;
 14:     11     (void) argv;
 15:     12
 16:     13     int i = 0;
 17:     14     while (environ[i]) {
 18:     15        printf("%s\n", environ[i++]);
 19:     16     }
 20:     17     return 0;
 21:     18  }
 22:     19
 23: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 24: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 environ.c -o environ
 25: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
    1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    2: err2.c
    3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    4:      1  // $Id: err2.c,v 1.5 2014-04-23 21:19:13-07 - - $
    5:      2
    6:      3  #include <stdio.h>
    7:      4  #include <string.h>
    8:      5  int main (void) {
    9:      6     char *p = 20;
   10:      7     int c = strcmp (p, "foo");
   11:      8     printf ("%d\n", c);
   12:      9     return 0;
   13:     10  }
   14: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   15: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 err2.c -o err2
   16: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   17: err2.c: In function â\200\230mainâ\200\231:
   18: err2.c:6:14: warning: initialization makes pointer from integer without
a cast [enabled by default]
   19:     char *p = 20;
   20:              ^
```

```
    1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    2: err.c
    3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    4:      1  // $Id: err.c,v 1.5 2014-04-23 21:19:13-07 - - $
    5:      2
    6:      3  #include <stdio.h>
    7:      4  void foo() {
    8:      5     int x = 20;
    9:      6     int *p = &x;
   10:      7  }
   11:      8  void bar() {
   12:      9     int a;
   13:     10     int *p;
   14:     11     printf ("%d %p\n", a, p);
   15:     12  }
   16:     13  int main() {
   17:     14     foo();
   18:     15     bar();
   19:     16  }
   20: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   21: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 err.c -o err
   22: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   23: err.c: In function â\200\230fooâ\200\231:
   24: err.c:6:9: warning: unused variable â\200\230pâ\200\231 [-Wunused-variab
le]
   25:     int *p = &x;
   26:          ^
   27: err.c: In function â\200\230barâ\200\231:
   28: err.c:11:11: warning: â\200\230aâ\200\231 is used uninitialized in this
function [-Wuninitialized]
   29:     printf ("%d %p\n", a, p);
   30:             ^
   31: err.c:11:11: warning: â\200\230pâ\200\231 is used uninitialized in this
function [-Wuninitialized]
```

```
    1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    2: list1.c
    3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    4:      1  // $Id: list1.c,v 1.3 2013-02-08 18:00:41-08 - - $
    5:      2
    6:      3  #include <assert.h>
    7:      4  #include <stdio.h>
    8:      5
    9:      6  typedef struct node node;
   10:      7  struct node {
   11:      8     char *word;
   12:      9     node *link;
   13:     10  };
   14:     11
   15:     12  int main (int argc, char **argv) {
   16:     13     node *head;
   17:     14     for (int argi = 0; argi < 5; ++argi) {
   18:     15        node *node = malloc (sizeof (struct node));
   19:     16        assert (node != NULL);
   20:     17        node->word = argv[argi];
   21:     18        node->link = head;
   22:     19        head = node;
   23:     20     }
   24:     21     for (node *curr = head; curr->link != NULL; curr = curr->link
) {
   25:     22        printf ("%p->node {word=%p->[%s], link=%p}\n",
   26:     23                curr, curr->word, curr->word, curr->link);
   27:     24     }
   28:     25     return 9;
   29:     26  }
   30: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   31: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 list1.c -o list1
   32: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   33: list1.c: In function â\200\230mainâ\200\231:
   34: list1.c:15:7: warning: implicit declaration of function â\200\230mallocâ
\200\231 [-Wimplicit-function-declaration]
   35:        node *node = malloc (sizeof (struct node));
   36:        ^
   37: list1.c:15:20: warning: incompatible implicit declaration of built-in fu
nction â\200\230mallocâ\200\231 [enabled by default]
   38:        node *node = malloc (sizeof (struct node));
   39:                     ^
   40: list1.c:12:15: warning: unused parameter â\200\230argcâ\200\231 [-Wunuse
d-parameter]
   41:  int main (int argc, char **argv) {
   42:                ^
```

```
    1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    2: list2.c
    3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    4:      1  // $Id: list2.c,v 1.2 2014-05-04 18:29:20-07 - - $
    5:      2
    6:      3  #include <assert.h>
    7:      4  #include <stdio.h>
    8:      5  #include <stdlib.h>
    9:      6
   10:      7  typedef struct node node;
   11:      8  struct node {
   12:      9     char *word;
   13:     10     node *link;
   14:     11  };
   15:     12
   16:     13  int main (int argc, char **argv) {
   17:     14     (void) argc; // Fix unused variable warning
   18:     15     node *head;
   19:     16     for (int argi = 0; argi < 5; ++argi) {
   20:     17        node *node = malloc (sizeof (struct node));
   21:     18        assert (node != NULL);
   22:     19        node->word = argv[argi];
   23:     20        node->link = head;
   24:     21        head = node;
   25:     22     }
   26:     23     for (node *curr = head; curr->link != NULL; curr = curr->link
) {
   27:     24        printf ("%p->node {word=%p->[%s], link=%p}\n",
   28:     25                curr, curr->word, curr->word, curr->link);
   29:     26     }
   30:     27     return 9;
   31:     28  }
   32: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   33: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 list2.c -o list2
   34: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
  1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  2: list3.c
  3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  4:      1  // $Id: list3.c,v 1.13 2014-05-04 20:07:50-07 - - $
  5:      2
  6:      3  #include <assert.h>
  7:      4  #include <stdio.h>
  8:      5  #include <stdlib.h>
  9:      6
 10:      7  typedef struct node node;
 11:      8  struct node {
 12:      9     char *word;
 13:     10     node *link;
 14:     11  };
 15:     12
 16:     13  int main (int argc, char **argv) {
 17:     14     node *head = NULL;
 18:     15     for (int argi = 1; argi < argc; ++argi) {
 19:     16        node *node = malloc (sizeof (struct node));
 20:     17        assert (node != NULL);
 21:     18        node->word = argv[argi];
 22:     19        node->link = head;
 23:     20        head = node;
 24:     21     }
 25:     22     for (node *curr = head; curr != NULL; curr = curr->link) {
 26:     23        printf ("%p->node {word=%p->[%s], link=%p}\n",
 27:     24                curr, curr->word, curr->word, curr->link);
 28:     25     }
 29:     26     return 9;
 30:     27  }
 31: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 32: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 list3.c -o list3
 33: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
  1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  2: list4.c
  3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  4:      1  // $Id: list4.c,v 1.3 2014-05-07 21:42:09-07 - - $
  5:      2
  6:      3  #include <assert.h>
  7:      4  #include <stdio.h>
  8:      5  #include <stdlib.h>
  9:      6
 10:      7  typedef struct node node;
 11:      8  struct node {
 12:      9     char *word;
 13:     10     node *link;
 14:     11  };
 15:     12
 16:     13  int main (int argc, char **argv) {
 17:     14     node *head = NULL;
 18:     15     for (int argi = 1; argi < argc; ++argi) {
 19:     16        node *node = malloc (sizeof (struct node));
 20:     17        assert (node != NULL);
 21:     18        node->word = argv[argi];
 22:     19        node->link = head;
 23:     20        head = node;
 24:     21     }
 25:     22     for (node *curr = head; curr != NULL; curr = curr->link) {
 26:     23        printf ("%p->node {word=%p->[%s], link=%p}\n",
 27:     24                curr, curr->word, curr->word, curr->link);
 28:     25     }
 29:     26     node *curr = head;
 30:     27     while (curr != NULL) {
 31:     28        node *tmp = curr;
 32:     29        curr = curr->link;
 33:     30        free (tmp);
 34:     31     }
 35:     32     head = NULL;
 36:     33     return EXIT_SUCCESS;
 37:     34  }
 38: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 39: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 list4.c -o list4
 40: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
    1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    2: malloc.c
    3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    4:      1  // $Id: malloc.c,v 1.2 2013-02-08 18:01:12-08 - - $
    5:      2
    6:      3  #include <stdio.h>
    7:      4  #include <stdlib.h>
    8:      5
    9:      6  typedef struct node node;
   10:      7  struct node {
   11:      8     int value;
   12:      9     node *link;
   13:     10  };
   14:     11
   15:     12  int main (int argc, char **argv) {
   16:     13     node *ptr = malloc (sizeof (struct node));
   17:     14     ptr = malloc (sizeof (node));
   18:     15     ptr->value = 6;
   19:     16     ptr->link = NULL;
   20:     17     printf ("%p-> {%d, %p}\n", ptr, ptr->value, ptr->link);
   21:     18     free (ptr);
   22:     19     return EXIT_SUCCESS;
   23:     20  }
   24: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   25: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 malloc.c -o malloc
   26: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   27: malloc.c: In function â\200\230mainâ\200\231:
   28: malloc.c:12:15: warning: unused parameter â\200\230argcâ\200\231 [-Wunus
ed-parameter]
   29:  int main (int argc, char **argv) {
   30:               ^
   31: malloc.c:12:28: warning: unused parameter â\200\230argvâ\200\231 [-Wunus
ed-parameter]
   32:  int main (int argc, char **argv) {
   33:                            ^
```

```
    1: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    2: uninit.c
    3: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    4:      1  // $Id: uninit.c,v 1.3 2014-02-18 15:08:14-08 - - $
    5:      2
    6:      3  #include <stdio.h>
    7:      4  #include <stdlib.h>
    8:      5
    9:      6  int main (int argc, char **argv) {
   10:      7     int *pointer;
   11:      8     printf ("pointer=%p\n", pointer);
   12:      9     printf ("*pointer=%d\n", *pointer);
   13:     10     int foo;
   14:     11     printf ("foo=%d\n", foo);
   15:     12  }
   16: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   17: Errors: gcc -g -O0 -Wall -Wextra -std=gnu99 uninit.c -o uninit
   18: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   19: uninit.c: In function â\200\230mainâ\200\231:
   20: uninit.c:6:15: warning: unused parameter â\200\230argcâ\200\231 [-Wunuse
d-parameter]
   21:  int main (int argc, char **argv) {
   22:                ^
   23: uninit.c:6:28: warning: unused parameter â\200\230argvâ\200\231 [-Wunuse
d-parameter]
   24:  int main (int argc, char **argv) {
   25:                             ^
   26: uninit.c:8:11: warning: â\200\230pointerâ\200\231 is used uninitialized
in this function [-Wuninitialized]
   27:     printf ("pointer=%p\n", pointer);
   28:             ^
   29: uninit.c:11:11: warning: â\200\230fooâ\200\231 is used uninitialized in
this function [-Wuninitialized]
   30:     printf ("foo=%d\n", foo);
   31:             ^
```