

```
1: #!/bin/sh
2: # $Id: mk,v 1.9 2014-04-30 17:37:19-07 - - $
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=$PATH:/afs/cats.ucsc.edu/courses/cmps012b-wm/bin
9: GCC='gcc -g -O0 -Wall -Wextra -std=gnu99'
10:
11: for CSOURCE in *.c
12: do
13:     EXECBIN=${CSOURCE%.c}
14:     if [ "$CSOURCE" == "$EXECBIN" ]
15:     then
16:         echo $0: $CSOURCE: is not a C source file
17:     else
18:         echo CSOURCE=$CSOURCE EXECBIN=$EXECBIN
19:         set -x
20:         cid + $CSOURCE
21:         checksource $CSOURCE
22:         $GCC $CSOURCE -o $EXECBIN -lm 2>&1 | tee $CSOURCE.log 2>&1
23:         set +x
24:         echo =====
25:     fi
26: done
27:
```

```
1: #!/bin/sh
2: # $Id: mkps,v 1.1 2014-04-30 17:37:19-07 - - $
3: mk *.c
4: for file in *.c
5: do
6:     more $file $file.log >$file.lis </dev/null
7: done
8: mkpspdf Listing.ps mk* *.lis
```

```
1: ::::::::::::::
2: err.c
3: ::::::::::::::
4: // $Id: err.c,v 1.5 2014-04-23 21:19:13-07 - - $
5:
6: #include <stdio.h>
7: void foo() {
8:     int x = 20;
9:     int *p = &x;
10: }
11: void bar() {
12:     int a;
13:     int *p;
14:     printf ("%d %p\n", a, p);
15: }
16: int main() {
17:     foo();
18:     bar();
19: }
20: ::::::::::::::
21: err.c.log
22: ::::::::::::::
23: err.c: In function 'foo':
24: err.c:6:9: warning: unused variable 'p' [-Wunused-variable]
25:     int *p = &x;
26:         ^
27: err.c: In function 'bar':
28: err.c:11:11: warning: 'a' is used uninitialized in this function [-Wunin
ialized]
29:     printf ("%d %p\n", a, p);
30:         ^
31: err.c:11:11: warning: 'p' is used uninitialized in this function [-Wunin
ialized]
```

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

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

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

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