© 2016 - 2017 — Rev: 10 – 13-Oct-2017. Silakan mengubah, memperbanyak, serta mendistribusikan dokumen ini selama tidak menghapus ketentuan ini. URL: http://rms46.vlsm.org/2/199.pdf

1. **2016-1**

Lingkari atau beri silang huruf "**B**" jika betul, dan "**S**" jika salah.

```
001 /* (c) 2015-2016 Rahmat M. Samik-Ibrahim    *
002  * R: 05-Jun-2016  -- This is free software */
003
004 #include <stdio.h>
005 #include <string.h>
006 #include <unistd.h>
007 #include <fcntl.h>
008 #include <sys/types.h>
009 #include <sys/stat.h>
010
011 char *string = "ABCD\n";
012 void main() {
013    int   fileDescriptor;
014    close(STDOUT_FILENO);
015    fileDescriptor = open ("output.txt", O_RDWR|O_CREAT|O_TRUNC, 0644);
016    printf (        "%s", string);
017    write(fileDescriptor, string, strlen(string));
018 }
```

B / S Tanpa baris 004 - 009, program akan tetap dapat dikompilasi tanpa kesalahan (error).

B / S Pointer "string" (baris 011) merupakan variabel global.

B / S Deklarasi "void main()" (baris 12) artinya: tidak ada "passing argument" ke dalam fungsi main().

B / S Pada saat program dieksekusi, secara otomatis file descriptor dari streams stdin=0 (STDIN_FILENO), stdout=1 (STDOUT_FILENO), dan stderr=2 (STDERR_FILENO).

B / S Baris 14 akan menutup stream STDOUT_FILENO (1).

B / S Nilai "fileDescriptor" = 1 (baris 15), akibat baris no 14.

B / S Jika berkas "output.txt" tidak ada (baris 15), maka fungsi open() akan membuat berkas "output.txt" baru.

B / S Jika sudah ada berkas "output.txt" (baris 15), maka fungsi open() akan membuka berkas dengan mode menambah (append).

B / S Fungsi "printf()" (baris 16) akan menulis "ABCD\n" ke layar monitor.

Isi semula berkas "output.txt" ialah "XXXX\n"; maka setelah program dieksekusi akan berisi:

2. **2016-2**

```
001 /* (c) 2016 Rahmat M. Samik-Ibrahim -- This is free software
005  * Assume (&ptrchr is 0x7FFFEEDDCCBB, order of bytes: little-endian) */
009 #define LINES  3
010 #include <stdio.h>
012 void printeq(int lines) {
013    while (lines-- > 0 ) printf("= = ");
014    printf("\n");
015 }
017 void main(void) {
018    int             ii;
019    unsigned char   dummy  = 'a';
020    unsigned char*  ptrchr = &dummy;
022    printeq(LINES);
023    printf(" dummy:  %c\n", dummy);
024    printf("*ptrchr: %c\n", *ptrchr);
025    printeq(LINES);
026    printf("%p\n", &ptrchr);
027    printeq(LINES);
028    ptrchr = (char*) &ptrchr;
029    for (ii=0; ii<6; ii++) {
030       printf("%X ", *ptrchr);
031       ptrchr++;
032    }
033    putchar('\n');
034    printeq(LINES);
035 }
```

(a) Write down the output of this program

3. **2017-1**

C Programing

```
001 /*
002  * (c) 2017 Rahmat M. Samik-Ibrahim --
           This is free software
003  * REV00 Thu Mar 30 16:56:54 WIB 2017
004  * START Thu Mar 30 16:56:54 WIB 2017
005  *
006  * fd2=dup(fd1)  duplicates fd1 to fd2
007  * O_RDWR Open the file so that it can be read
               from and written to.
008  * O_TRUNC Initially clear all data from the file.
009  * O_CREAT If the file does not exist, create it.
010  */
011
012 #include <stdio.h>
013 #include <unistd.h>
014 #include <sys/types.h>
015 #include <sys/stat.h>
016 #include <fcntl.h>
017 #include <string.h>
018 #define FILE "uts2017-1.txt"
020 static char* str1 = "AABB\n";
021 static char* str2 = "CCDD\n";
022 static char* str3 = "EEFF\n";
023
024 void main(void) {
025    int fd1, fd2, fd3;
026    /* STDIN=0, STDOUT=1, STDERR=2, therefore
027       fd1, fd2, fd3  will be 3, 4, and 5 */
028    fd1 = open (FILE, O_TRUNC | O_RDWR | O_CREAT, 0644);
029    fd2 = open (FILE, O_TRUNC | O_RDWR | O_CREAT, 0644);
030    fd3 = dup(fd2);
031    printf("fd1=%d, fd2=%d, fd3=%d\n", fd1, fd2, fd3);
032    write(fd1, str1, strlen(str1));
033    write(fd2, str2, strlen(str2));
034    write(fd3, str3, strlen(str3));
035    close(fd1);
036    close(fd2);
037    close(fd3);
038 }
```

Program Output (Line 031):

Content of file "uts2017-1.txt"

C Programing

```
001 /*
002  * (c) 2017 Rahmat M. Samik-Ibrahim
           -- This is free software
003  * REV00 Thu Mar 30 18:27:30 WIB 2017
004  * START Thu Mar 30 18:27:30 WIB 2017
005  * INT is 32 bit little endian
006  * 41H='A'; 42H='B'; 43H='C"; 44H='D'
007  */
008
009 #include <stdio.h>
010 char chrary[]="ZZZZ ZZZZ ";
011 void main(void) {
012    char  chrvar = 'M';
013    int   intvar = 0x41424344;
014    int*  intptr = (int*) chrary;
015    printf("YY.     chrary=%p\n",  chrary);
016    printf("ZZ.     intprt=%p\n",  intptr);
017    printf("01.     chrvar=%c\n",  chrvar);
018    printf("02.    *chrary=%c\n", *chrary);
019    printf("03. str chrary=%s\n",  chrary);
020    *intptr      =  intvar;
021    printf("04. str chrary=%s\n",  chrary);
022 }
```

Program Output (Line: 015, 016, 017, 018, 019, 021):