

# **PRAKTIKUM SISTEM OPERASI**



**Disusun Oleh :**

**Fariz Taufiqul Hafidz**

**L200210192**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA  
TAHUN 2022/2023**

1. Membuat sebuah 'Child process' (Proses baru) dengan menggunakan system call 'fork'.

### Kode Program

```
C fork.c x
home > farizhafidz > Documents > Modul_8 > C fork.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/types.h>
5  int main() {
6      pid_t pid;
7      int x = 5;
8      pid = fork();
9      x++;
10     if (pid < 0)
11     {
12         printf("process creation error"); exit(-1);
13     }
14     else if (pid == 0)
15     {
16         printf("Child process:");
17         printf("\nProcess id is %d", getpid());
18         printf("\nValue of x is %d", x);
19         printf("\nProcess id of parent is %d\n\n", getpid());
20     }
21     else
22     {
23         printf("\nParent process:");
24         printf("\nProcess id is %d", getpid());
25         printf("\nValue of x is %d", x);
26         printf("\nProcess id of parent is %d\n\n", getpid());
27     }
28 }
```

### Terminal (Output)

```
Farizh
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ gcc fork.c
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ ./a.out

Parent process:
Process id is 5293
Value of x is 6
Process id of parent is 5293

Child process:
Process id is 5294
Value of x is 6
Process id of parent is 5294
```

2. Menghentikan sementara (block) proses parent sampai dengan proses child selesai, menggunakan perintah system call 'wait'.

### Kode Program

```
C wait.c X
home > farizhafidz > Documents > Modul_8 > C wait.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/types.h>
5  #include <sys/wait.h>
6  int main() {
7      int i, status;
8      pid_t pid;
9      pid = fork();
10
11     if (pid < 0) {
12         printf("\nPembuatan proses gagal\n");
13         exit(-1);
14     }
15     else if (pid > 0)
16     {
17         wait(NULL);
18         printf("\nParent starts\nNomor Genap:");
19         for ( i=2; i<=10; i+=2)
20             printf ("%3d",i);
21         printf("\nParents ends\n");
22     }
23     else if (pid == 0)
24     {
25         printf("\nChild starts\nNomor Ganjil:");
26         for ( i=1; i<10; i+=2)
27             printf ("%3d",i);
28         printf("\nChild ends\n");
29     }
30 }
```

### Terminal (Output)

```
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ gcc wait.c
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ ./a.out

Child starts
Nomor Ganjil:  1  3  5  7  9
Child ends

Parent starts
Nomor Genap:  2  4  6  8 10
Parents ends
```

3. Loading program yang dapat dieksekusi dalam sebuah 'child' proses menggunakan perintah system call 'exec'.

### Kode Program

```
C exec.c X
home > farizhafidz > Documents > Modul_8 > C exec.c
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <unistd.h>
4  #include <stdlib.h>
5  int main (int argc, char*argv[]) {
6
7      pid_t pid;
8      int i;
9
10     if (argc != 3)
11     {
12         printf ("\nInsufficient arguments to load program");
13         printf ("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
14     }
15
16     switch (pid = fork())
17     {
18     case -1:
19         printf ("Fork failed");
20         exit(-1);
21     case 0:
22         printf("Child process\n");
23         i = execl(argv[1], argv[2], 0);
24         if (i < 0)
25         {
26             printf ("%s program not loaded using exec system call\n", argv[2]);
27             exit(-1);
28         }
29     default:
30         wait(NULL);
31         printf("Child Terminated\n");
32         exit(0);
33     }
34 }
```

### Terminal (Output)

```
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ gcc exec.c
exec.c: In function 'main':
exec.c:23:9: warning: missing sentinel in function call [-Wformat=]
   23 |         i = execl(argv[1], argv[2], 0);
      |         ^
exec.c:30:9: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
   30 |         wait(NULL);
      |         ^~~~~
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ ./a.out /bin/ls ls
Child process
a.out dirlist.c exec.c fork.c stat.c wait.c
Child Terminated
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$
```

4. Menampilkan status file menggunakan perintah system call 'stat'.

### Kode Program

```
C stat.c x
home > Farizhafidz > Documents > Modul_8 > C stat.c
1  #include <stdio.h>
2  #include <sys/stat.h>
3  #include <stdlib.h>
4  #include <time.h>
5  int main (int argc, char*argv[]) {
6      struct stat
7      file; int n;
8
9      if (argc != 2)
10     {
11         printf ("Usage: ./a.out <filename>\n"); exit(-1);
12     }
13     if ((n = stat(argv[1], &file)) == -1)
14     {
15         perror(argv[1]);
16         exit(-1);
17     }
18     printf("User id : %d\n", file.st_uid);
19     printf("Group id : %d\n", file.st_gid);
20     printf("Block size : %ld\n", file.st_blksize);
21     printf("Block allocated : %ld\n", file.st_blocks);
22     printf("Inode no. : %ld\n", file.st_ino);
23     printf("Last accessed : %s", ctime(&(file.st_atime)));
24     printf("Last modified : %s", ctime(&(file.st_mtime)));
25     printf("File size : %ld bytes\n", file.st_size);
26     printf("No. of links : %ld\n", file.st_nlink);
27     printf("Permissions : ");
28     printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
29     printf( (file.st_mode & S_IRUSR) ? "r" : "-");
30     printf( (file.st_mode & S_IWUSR) ? "w" : "-");
31     printf( (file.st_mode & S_IXUSR) ? "x" : "-");
32     printf( (file.st_mode & S_IRGRP) ? "r" : "-");
33     printf( (file.st_mode & S_IWGRP) ? "w" : "-");
34     printf( (file.st_mode & S_IXGRP) ? "x" : "-");
35     printf( (file.st_mode & S_IROTH) ? "r" : "-");
36     printf( (file.st_mode & S_IWOTH) ? "w" : "-");
37     printf( (file.st_mode & S_IXOTH) ? "x" : "-");
38     printf("\n");
39     if (file.st_mode & S_IFREG)
40         printf ("File type : Regular\n");
41     if (file.st_mode & S_IFDIR)
42         printf ("File type : Directory\n");
43 }
```

### Terminal (Output)

```
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ gcc stat.c
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$ ./a.out stat.c
User id : 1000
Group id : 1000
Block size : 4096
Block allocated : 8
Inode no. : 788184
Last accessed : Thu Dec 15 13:11:12 2022
Last modified : Thu Dec 15 13:11:09 2022
File size : 1493 bytes
No. of links : 1
Permissions : -rw-rw-r--
File type : Regular
farizhafidz@farizhafidz-VirtualBox:~/Documents/Modul_8$
```

5. Menampilkan isi direktori menggunakan perintah system call 'readdir'.

### Kode Program

```
C dirlist.c X
home > farizhafidz > Documents > Modul_8 > C dirlist.c
1  #include <stdio.h>
2  #include <dirent.h>
3  #include <stdlib.h>
4  int main (int argc, char*argv[]) {
5      struct dirent *dptr;
6      DIR *dname;
7
8      if (argc != 2)
9      {
10         printf("Usage : ./a.out <dirname>\n");
11         exit(-1);
12     }
13     if((dname = opendir(argv[1])) == NULL)
14     {
15         perror(argv[1]);
16         exit(-1);
17     }
18     while(dptr = readdir (dname))
19         printf ("%s\n", dptr->d_name);
20
21     closedir(dname);
22
23 }
```

### Terminal (Output)

```
farizhafidz@farizhafidz-VirtualBox: ~/Documents/Modul_8$ gcc dirlist.c
farizhafidz@farizhafidz-VirtualBox: ~/Documents/Modul_8$ ./a.out Screenshots
Screenshot from 2022-12-08 20-24-53.png
..
Screenshot from 2022-12-08 20-03-50.png
Screenshot from 2022-12-08 10-37-02.png
Screenshot from 2022-12-08 20-24-19.png
Screenshot from 2022-12-08 10-36-19.png
Screenshot from 2022-12-08 19-42-59.png
.
farizhafidz@farizhafidz-VirtualBox: ~/Documents/Modul_8$
```