

Nama : Alfian pandu
NIM : L200180027
Kelas : A

MODUL 8

1. Membuat child “Child Process” dengan menggunakan System Call“Fork”

The screenshot shows two terminal windows side-by-side. The left terminal window displays the source code for `fork.c`, which includes #include directives for `<stdio.h>`, `<stdlib.h>`, `<unistd.h>`, and `<sys/types.h>`. The `main()` function contains logic to check if it's a child process (pid < 0), a parent process (pid == 0), or a shell (else). It then prints the process ID and value of variable `x` for each case. The right terminal window shows the compilation of `fork.c` using `gcc`, followed by the execution of the generated binary `a.out`. The output shows the parent process (PID 1537) and the child process (PID 1538) both printing their respective process IDs and the value of `x`.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
int main() {
    pid_t pid;
    int x = 5;
    pid = fork();
    x++;
    if (pid < 0)
    {
        printf("Process creation error"); exit(-1);
    }
    else if (pid == 0)
    {
        printf("Child process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of parent is %d\n\n", getppid());
    }
    else
    {
        printf("\nParent process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of shell is %d\n", getppid());
    }
}
```

```
fork.c:25:10: warning: too many arguments for format [-Wformat-extra-args]
    printf("\nValue of x is %", x);
               ^
fork.c: In function 'main':
fork.c:19:10: warning: unknown escape sequence: '\P'
    printf("Process id of parent is %d\n\n", getppid());
               ^
fork.c:25:27: warning: spurious trailing '%' in format [-Wformat=]
    printf("\nValue of x is %", x);
               ^
fork.c:25:10: warning: too many arguments for format [-Wformat-extra-args]
    printf("\nValue of x is %", x);
               ^
kayon-pc16@kayon-pc16:~/Desktop$ gcc fork.c
fork.c: In function 'main':
fork.c:19:10: warning: unknown escape sequence: '\P'
    printf("Process id of parent is %d\n\n", getppid());
               ^
fork.c:25:27: warning: spurious trailing '%' in format [-Wformat=]
    printf("\nValue of x is %", x);
               ^
fork.c:25:10: warning: too many arguments for format [-Wformat-extra-args]
    printf("\nValue of x is %", x);
               ^
kayon-pc16@kayon-pc16:~/Desktop$ ./a.out
Parent process:
Process id is 1537
Value of x is 6
Process id of shell is 1481
Child process:
Process id is 1538
Value of x is 6
Process id of parent is 1537
```

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

The screenshot shows a Linux desktop environment with two terminal windows and a nano editor window.

The left terminal window displays the source code for a C program named `wait.c`. The code uses the `fork()` function to create a child process. The parent process then calls `wait(NULL)` to block and wait for the child process to terminate. The child process prints odd numbers from 1 to 9. The parent process prints even numbers from 2 to 10. The code is as follows:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
int main() {
    int i, status;
    pid_t pid;
    pid = fork();

    if (pid < 0) {
        printf("\nPembuatan proses gagal\n");
        exit(-1);
    }
    else if (pid > 0)
    {
        wait(NULL);
        printf ("\nParent starts\nNomor Genap:");
        for (i=2;i<=10;i+=2)
            printf ("%3d",i);
        printf("\nParent ends\n");
    }
    else if (pid == 0)
    {
        printf("Child starts\nNomor Ganjil:");
        for (i=1;i<10;i+=2)
            printf ("%3d",i);
        printf ("\nChild ends\n");
    }
}
```

The right terminal window shows the execution of the program. It starts with `gcc wait.c` and then runs the executable `./a.out`. The output shows the parent process printing even numbers from 2 to 10, and the child process printing odd numbers from 1 to 9.

```
kayon-pc16@kayon-pc16:~/Desktop$ gcc wait.c
kayon-pc16@kayon-pc16:~/Desktop$ ./a.out
Child starts
Nomor Ganjil: 1 3 5 7 9
Child ends

Parent starts
Nomor Genap: 2 4 6 8 10
Parent ends
kayon-pc16@kayon-pc16:~/Desktop$
```

3. Loading Program yang dapat dieksekusi dalam sebuah "Child" Proses menggunakan perintah system call "exec"

```

Terminal - kayon-pc25@kayon-pc25: ~
File Edit View Terminal Tabs Help
root@kayon-pc25: /home/kay... x kayon-pc25@kayon-pc25: ~ x root@kayon-pc25: /home/kay... x kayon-pc25@kayon-pc25: ~ x root@kayon-pc25: /home/kay... x kayon-pc25@kayon-pc25: ~ x
kayon-pc25@kayon-pc25: ~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char*argv[])
^
exec.c: In function 'main':
exec.c:23:3: warning: missing sentinel in function call [-Wformat]
    i = exec(argv[1], argv[2], 0);
^
exec.c:30:3: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
    wait(NULL);
^
main
kayon-pc25@kayon-pc25: ~$ ./a.out /bin/ls ls
child process
a.out Desktop Documents Downloads exec.c fork.c infol200180110.sh.save infol200180110.sh.save.1 Music Pictures Public Templates Videos wait.c
Child Terminated
kayon-pc25@kayon-pc25: ~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char*argv[])
^
exec.c: In function 'main':
exec.c:20:3: warning: missing sentinel in function call [-Wformat]
    i = exec(argv[1], argv[2], 0);
^
exec.c:27:3: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
    wait(NULL);
^
main
kayon-pc25@kayon-pc25: ~$ gcc exec.c
exec.c:5:10: fatal error: sys/wait.c: No such file or directory
 #include <sys/wait.c>
^
compilation terminated.
kayon-pc25@kayon-pc25: ~$ gcc exec.c
exec.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char*argv[])
^
kayon-pc25@kayon-pc25: ~$ ./a.out /bin/ls ls
child process
a.out Desktop Documents Downloads exec.c fork.c infol200180110.sh.save infol200180110.sh.save.1 Music Pictures Public Templates Videos wait.c
Child Terminated
kayon-pc25@kayon-pc25: ~

```

```

Terminal - root@kayon-pc25:/home/kayon-pc25
File Edit View Terminal Tabs Help
root@kayon-pc25: /home/kay... x kayon-pc25@kayon-pc25: ~ x root@kayon-pc25: /home/kay... x kayon-pc25@kayon-pc25: ~ x root@kayon-pc25: /home/kay... x kayon-pc25@kayon-pc25: ~ x
GNU nano 2.9.3
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
int main(int argc, char*argv[])
{
    pid_t pid;
    int i;
    if (argc != 3)
    {
        printf("\nInsufficient arguments to load program");
        printf("\nUsage: ./a.out <path> <cmd>\n");
        exit(-1);
    }
    switch(pid = fork())
    {
    case -1:
        printf("Fork failed");
        exit(-1);
    case 0:
        printf("Child process\n");
        i = exec(argv[1], argv[2], NULL);
        if (i < 0)
        {
            printf("%s program not loaded using exec system call\n", argv[2]);
            exit(-1);
        }
    default:
        wait(NULL);
        printf("Child Terminated\n");
        exit(-1);
    }
}

```

4. Menambahkan status file menggunakan perintah system call "stat"

The image shows three vertically stacked terminal windows from a Linux environment, likely a Kali Linux distribution, illustrating the step-by-step development of a C program named `stat.c`.

Top Terminal: Displays the initial state of the C code, which includes basic file handling and printing of file statistics.

```
GNU nano 2.9.3
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>

int main(int argc, char*argv[]) {
    struct stat
    file; int n;
    if (argc != 2)
    {
        printf("Usage: ./a.out <filename>\n"); exit(-1);
    }
    if ((n = stat(argv[1], &file)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    printf("User id : %d\n", file.st_uid);
    printf("Group id : %d\n", file.st_gid);
    printf("Block size : %d\n", file.st_blksize);
    printf("Blocks allocated : %d\n", file.st_blocks);
    printf("Inode no. : %d\n", file.st_ino);
    printf("Last accessed : %s", ctime(&(file.st_atime)));
    printf("Last modified : %s", ctime(&(file.st_mtime)));
    printf("File size : %d bytes\n", file.st_size);
    printf("No. of links : %d\n", file.st_nlink);
    printf("Permission : ");
}
```

Middle Terminal: Shows the addition of a conditional block to handle different file types based on the `S_ISDIR` macro.

```
GNU nano 2.9.3
    perror(argv[1]);
    exit(-1);
}
printf("User id : %d\n", file.st_uid);
printf("Group id : %d\n", file.st_gid);
printf("Block size : %d\n", file.st_blksize);
printf("Blocks allocated : %d\n", file.st_blocks);
printf("Inode no. : %d\n", file.st_ino);
printf("Last accessed : %s", ctime(&(file.st_atime)));
printf("Last modified : %s", ctime(&(file.st_mtime)));
printf("File size : %d bytes\n", file.st_size);
printf("No. of links : %d\n", file.st_nlink);
printf("Permission : ");
if(S_ISDIR(file.st_mode)) ? "d" : "-";
if(file.st_mode & S_IRUSR) ? "r" : "-";
if(file.st_mode & S_IWUSR) ? "w" : "-";
if(file.st_mode & S_IXUSR) ? "x" : "-";
if(file.st_mode & S_IRGRP) ? "r" : "-";
if(file.st_mode & S_IWGRP) ? "w" : "-";
if(file.st_mode & S_IXGRP) ? "x" : "-";
if(file.st_mode & S_IROTH) ? "r" : "-";
if(file.st_mode & S_IWOTH) ? "w" : "-";
if(file.st_mode & S_IXOTH) ? "x" : "-";
printf("\n");
if(file.st_mode & S_IFREG)
    printf("File type : Regular\n");
else if(file.st_mode & S_IFDIR)
    printf("File type : Directory\n");
else
    printf("File type : Unknown\n");
}
```

Bottom Terminal: Shows the final output of the compiled program `a.out` when run with a file name argument.

```
File type : Regular
fltria@fltria-VirtualBox:~$ gcc stat.c
fltria@fltria-VirtualBox:~$ ./a.out
Usage: ./a.out <filename>
fltria@fltria-VirtualBox:~$ gcc stat.c
fltria@fltria-VirtualBox:~$ ./a.out stat.c
User id : 0
Group id : 0
Block size : 4096
Blocks allocated : 8
Inode no. : 555318
Last accessed : Thu Nov 21 21:09:24 2019
Last modified : Thu Nov 21 21:09:16 2019
File size : 1375 bytes
No. of links : 1
Permission : -rw-r--r--
File type : Regular
fltria@fltria-VirtualBox:~$ gcc stat.c
fltria@fltria-VirtualBox:~$ ./a.out stat.c
User id : 0
Group id : 0
Block size : 4096
Blocks allocated : 8
Inode no. : 555318
Last accessed : Thu Nov 21 21:09:24 2019
Last modified : Thu Nov 21 21:09:16 2019
File size : 1375 bytes
No. of links : 1
Permission : -rw-r--r--
File type : Regular
fltria@fltria-VirtualBox:~$
```

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

GNU nano 2.9.3 dirlist.c

```
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>
int main(int argc, char*argv[]){
    struct dirent *dptr;
    DIR *dname;
    if (argc != 2)
    {
        printf("Usage: ./a.out <dirname>\n");
        exit(-1);
    }
    if((dname = opendir(argv[1])) == NULL)
    {
        perror(argv[1]);
        exit(-1);
    }
    while(dptr=readdir(dname))
        printf("%s\n", dptr->d_name);

    closedir(dname);
}
```

```
fitria@fitria-VirtualBox:~$ gcc dirlist.c
fitria@fitria-VirtualBox:~$ ./a.out dirlist.c
User id : 0
Group id : 0
Block size : 4096
Blocks allocated : 8
Inode no. : 530210
Last accessed : Thu Nov 21 21:17:00 2019
Last modified : Thu Nov 21 21:16:54 2019
File size : 360 bytes
No. of links : 1
Permission : -rw-r--r--
File type : Regular
fitria@fitria-VirtualBox:~$ ./a.out Downloads
.
.
.
google-chrome-stable_current_amd64.deb
fitria@fitria-VirtualBox:~$ ./a.out Pictures
Screenshot from 2019-11-09 14-45-38.png
.
.
.
Screenshot from 2019-11-09 14-48-13.png
Screenshot from 2019-11-09 14-48-33.png
fitria@fitria-VirtualBox:~$
```