

Project#1:Multi-process and IPC

Course: Operatin System

Name: Fei Gao

Structure

```
eachProcs = numFiles/numProcs;
extraFiles = numFiles - (eachProcs * numProcs);
procsId = 0;

// assign exact division files to each child process
for ( k = 0; k < numProcs; k++){
    count[k] = eachProcs;
}

// assign extra files to child processes
while(extraFiles != 0){
    count[procsId] = count[procsId] + 1;
    procsId++;
    extraFiles = extraFiles - 1;
}
```

This part is used to declare how many files will be assigned to each child process. The quantity of each child process should read is stored in `count[]`.

```

for ( m = 0; m < numProcs; m++){
    currentFiles = currentFiles + count[m];
    if (m == 0)
        printf("Child Proc %d, read %d files (0 ~ %d)\n",m,count[m],
            count[m] - 1);
    else
        printf("Child Proc %d, read %d files (%d ~ %d)\n",m,count[m],
            currentFiles-count[m],currentFiles-1);
}

```

This part is used to assign exactly which file should be read by which child process.

```

if( i != 0 ){
    numText = numText + count[i-1];
}

```

In the loop, this part is to update the number of the first file that each child process should read.

```

sprintf(filename, "%s/text.%02d", FILEPATH, numText);
numText = numText + 1;

```

In the inner loop, this part is to read each file by each child process.

```

for(i = 0; i < numProcs; i++)
{
    waitpid(pid[i], &wstatus, 0);
    if(WIFEXITED(wstatus)) printf("The child process %d terminated
        normally. The Exit status %d\n", pid[i], WEXITSTATUS(wstatus));
    if(WIFSIGNALED(wstatus)) printf("The child process %d terminated
        by a signal %d.\n", pid[i], WTERMSIG(wstatus));
}

```

In the end, using `waitpid` to let the parent to wait child processes terminated. Also we can use `kill -9 pid` to terminate it.

IPC

```
if (pipe(pipefd) == -1){  
    perror("Failed to create pipe\n");  
    exit (EXIT_FAILURE);  
}
```

This part is to check the child process create successfully or not.

- Firstly, using `write (pipefd[1], &total, sizeof(count_t));` to put the initial value in pipe.
- Then, in child process, using `write (pipefd[1], &total, sizeof(count_t));` to put the value in pipe.
- Finally, in parent process, using `read (pipefd[0], &total, sizeof(count_t));` to get the value from the pipe.

The important thing is, after using read/write ends, do not forget to close.

Result

Here is the output of wc_mul.c

```
[~bash-4.1$ ./wc_mul 10 4
counting 10 files in 4 processes.
Child Proc 0, read 3 files (0 ~ 2)
Child Proc 1, read 3 files (3 ~ 5)
Child Proc 2, read 2 files (6 ~ 7)
Child Proc 3, read 2 files (8 ~ 9)
[pid 23735] read: /tmp/CSCI4730/books/text.00
[pid 23736] read: /tmp/CSCI4730/books/text.03
[pid 23737] read: /tmp/CSCI4730/books/text.06
[pid 23738] read: /tmp/CSCI4730/books/text.08
[pid 23736] read: /tmp/CSCI4730/books/text.04
[pid 23738] read: /tmp/CSCI4730/books/text.09
[pid 23735] read: /tmp/CSCI4730/books/text.01
[pid 23737] read: /tmp/CSCI4730/books/text.07
[pid 23737] send the result to the parent 23734.
[pid 23738] send the result to the parent 23734.
[pid 23735] read: /tmp/CSCI4730/books/text.02
[pid 23736] read: /tmp/CSCI4730/books/text.05
[pid 23735] send the result to the parent 23734.
The child process 23735 terminated normally. The Exit status 0
[pid 23736] send the result to the parent 23734.
The child process 23736 terminated normally. The Exit status 0
The child process 23737 terminated normally. The Exit status 0
The child process 23738 terminated normally. The Exit status 0
```

```
=====
Total Lines : 16177972
Total Words : 151538006
Total Characters : 665714062
=====
```

Here is the output of wc_mul which a child process is terminated by signal

```
[~bash-4.1$ ./wc_mul 10 4
counting 10 files in 4 processes.
Child Proc 0, read 3 files (0 ~ 2)
Child Proc 1, read 3 files (3 ~ 5)
Child Proc 2, read 2 files (6 ~ 7)
Child Proc 3, read 2 files (8 ~ 9)
[pid 26092] read: /tmp/CSCI4730/books/text.00
[pid 26093] read: /tmp/CSCI4730/books/text.03
[pid 26094] read: /tmp/CSCI4730/books/text.06
[pid 26095] read: /tmp/CSCI4730/books/text.08
The child process 26092 terminated by a signal 9.
[pid 26093] read: /tmp/CSCI4730/books/text.04
[pid 26095] read: /tmp/CSCI4730/books/text.09
[pid 26094] read: /tmp/CSCI4730/books/text.07
[pid 26095] send the result to the parent 26091.
[pid 26094] send the result to the parent 26091.
[pid 26093] read: /tmp/CSCI4730/books/text.05
[pid 26093] send the result to the parent 26091.
The child process 26093 terminated normally. The Exit status 0
The child process 26094 terminated normally. The Exit status 0
The child process 26095 terminated normally. The Exit status 0
```


=====

```
Total Lines : 10492951
Total Words : 98104301
Total Characters : 432770315
```

=====


Here is the time of wc

real	0m11.046s
user	0m10.686s
sys	0m0.362s



Here is the time of wc_mul

real	0m3.878s
user	0m10.372s
sys	0m0.349s



Obviously, wc_mul saves time.