

## OS Practical 4 :-

The screenshot shows the MSYS2 nano editor window. The file being edited is named 'practical4.c'. The code implements a round-robin scheduling algorithm with three processes (P1, P2, P3) over 6 time units. It calculates waiting times and context switches.

```
MSYS2 MSYS Shell
GNU nano 8.7 practical4.c
#include <stdio.h>

int main() {
    int n = 3;
    int arrival[] = {0, 2, 6};
    int burst[] = {10, 20, 30};
    int remaining[] = {10, 20, 30};
    int waiting[3] = {0, 0, 0};
    int completion[3] = {0, 0, 0};

    int time = 0, completed = 0;
    int prev = -1;
    int contextSwitches = 0;

    while (completed < n) {
        int shortest = -1;
        int min = 1e9;

        for (int i = 0; i < n; i++) {
            if (arrival[i] <= time && remaining[i] > 0 && remaining[i] < min) {
                min = remaining[i];
                shortest = i;
            }
        }

        if (shortest == -1) {
            time++;
            continue;
        }

        if (prev != -1 && prev != shortest)
            contextSwitches++;

        prev = shortest;
        remaining[shortest]--;
        time++;

        if (remaining[shortest] == 0) {
            completed++;
            completion[shortest] = time;
        }
    }

    int totalWait = 0;
    printf("Process Waiting Time\n");
    for (int i = 0; i < n; i++) {
        waiting[i] = completion[i] - arrival[i] - burst[i];
        totalWait += waiting[i];
        printf("%d ns\n", i + 1, waiting[i]);
    }

    printf("\nTotal Context switches = %d", contextSwitches);
    printf("\nAverage waiting Time = %.2f ns\n", (float)totalWait / n);
}

return 0;
}
```

At the bottom of the window, there is a toolbar with various text editing commands:

- Help
- Exit
- Write Out
- Read File
- Where Is
- Replace
- Cut
- Paste
- Execute
- Justify
- Location
- Go To Line
- Undo
- Redo
- Set Mark
- Copy
- To Bracket
- Where Was
- Previous
- Next
- Back
- Forward
- Prev Word
- Next Word

The screenshot shows the MSYS2 UCRT64 Shell window. The user has run the compiled program 'practical4' and its output is displayed.

```
barkha-dev:~$ nano practical4.c
barkha-dev:~$ gcc practical4.c -o practical4
barkha-dev:~$ ./practical4
Process Waiting Time
P1      0 ns
P2      8 ns
P3     24 ns

Total Context switches = 2
Average waiting Time = 10.67 ns
barkha-dev:~$
```

