|  |
| --- |
|  |
| 시스템 프로그래밍 |
|  |
| Assignment 6 |

**1분반**

2010136017

김덕균

시스템 프로그래밍

Assignment 6

# pmon.c 소스 및 주석

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <signal.h>

int getPID(char \*process\_name){

int pid = -1;

char cmd\_result[1024];

FILE \*fp;

//문자열 만들기

sprintf(cmd\_result,"ps -eaf | grep -w %s | grep -v grep | awk '{print $2}'",process\_name);

//해당 문자열 실행

fp = popen(cmd\_result,"r");

if(fp != NULL){

//pid 얻기

fseek(fp,0,SEEK\_SET);

fscanf(fp,"%d",&pid);

fclose(fp);

}

return pid;

}

void check\_process(){

//pid 값 얻어오기

int pid = getPID("ptest");

if(pid == -1){

printf("not existed\n");

}else{

printf("running\n");

}

//5초에 한번씩 알람

alarm(5);

}

int main(int argc,char\* argv[]){

char command=0;

int pid;

//프로세스 체크 함수 알람에 등록

signal(SIGALRM,check\_process);

alarm(5);

while(command != 'q'){

printf(">>");

scanf("%c",&command);

//버퍼 비우기

while(getchar()!='\n');

pid = getPID("ptest");

switch(command){

case 'Q':

case 'q':

printf("프로그램 종료\n");

command='q';

break;

case 'K':

case 'k':

if(pid != -1){

kill(pid,SIGKILL);

}else{

printf("not existed\n");

}

break;

case 'S':

case 's':

if(pid != -1){

printf("already running\n");

}else{

system("./ptest &");

}

break;

case 'R':

case 'r':

if(pid != -1){

kill(pid,SIGKILL);

}else{

printf("newly started\n");

}

system("./ptest &");

}

}

}

# Ptest.c 소스 및 주석

#include <stdio.h>

#include <unistd.h>

#include <signal.h>

#include <time.h>

void what\_time(){

struct tm \*tm\_ptr;

time\_t the\_time;

(void)time(&the\_time);

tm\_ptr = localtime(&the\_time);

printf("%d/%d/%d %d:%d:%d\n",

tm\_ptr->tm\_year+1900,tm\_ptr->tm\_mon+1,tm\_ptr->tm\_mday,

tm\_ptr->tm\_hour,tm\_ptr->tm\_min,tm\_ptr->tm\_sec);

}

int main(int argc,char\* argv[]){

//알람 시그널에 시간 출력 함수 등록

signal(SIGALRM,what\_time);

what\_time();

while(1){

alarm(2);

pause();

}

return 0;

}