

Date: 22-10-21

→ WAP To deliver SIGALRM signal to a long running process
after 10 seconds do the same repeatedly like after 1sec, 8sec,
7sec like this when number of seconds reaching to zero process

Should terminate.

→ WAP To find the current action of the signal?

main() int n;
signal(2, SIG-IGN.)
scanf("%d", &n) ⇒ ①

2 → SIG-IGN - ignore

main()
{

 fat n;
 void (*ptr)(int)

 signal (n, SIG_IGN);

 printf("enter the signal no %d");

 scanf("%d", &n);

ptr = signal (n, ISR);

 signal (n, ptr);

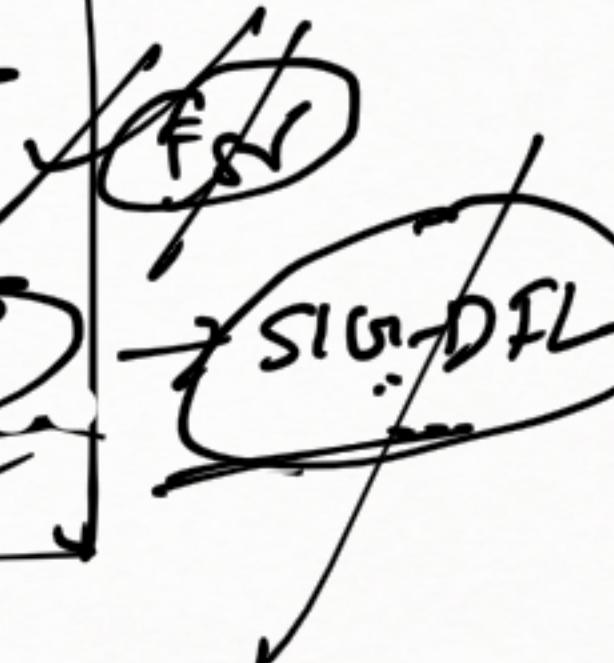
 if (ptr == SIG_DFL)

 ptr = "Default";

 else if (ptr == SIG_IGN)

 ptr = "ignore - [n]";

signo	action
1	Default
2	ignore
3	Default
4	Default



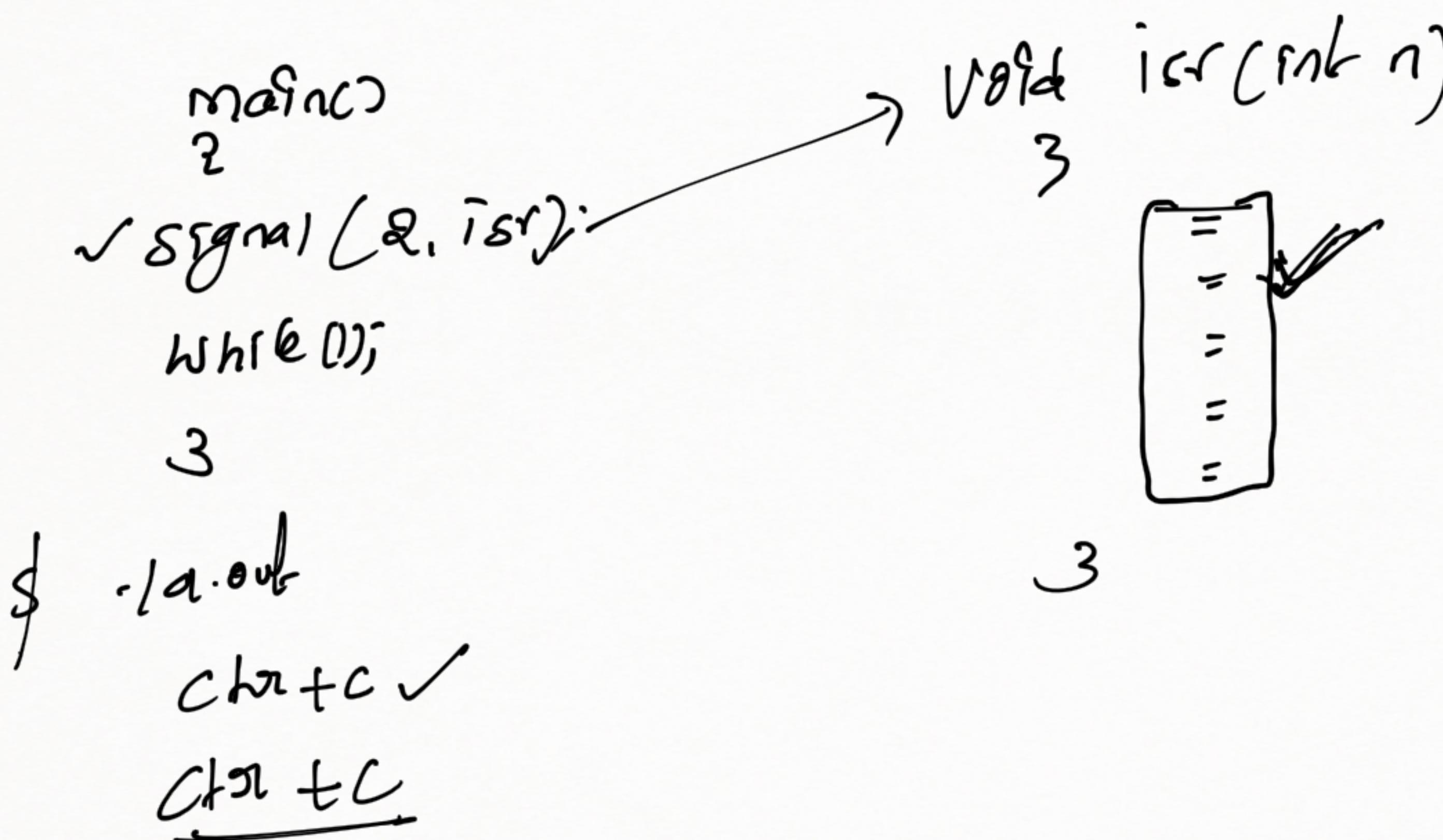
isn't SIG_DFL

→ using signal function return type possible to find the current action of signal

else
ptr ("ISR[n]");

→ To get the current action set a new returns current action.

→ observe that when one signal ISR is under execution if the same signal again raised it is allowed (cont.)



Conclusion For example if signum & ISR is under execution at that time if the same signal raised it is blocked. but other signals (signo 3) are allowed

Sigaction
part sigaction (int signum, const struct sigaction *act, struct sigaction *old)

(i) sigaction (signum, &v, 0) Struct sigaction v
In this case user want to install a new action to the signal but user dont want to cancel the current action

(i) `sigaction(num, &v);` struct sigaction v;
User want to collect the current action of the signal but don't want to install new action for the signal

(ii) `sigaction(num, &v1, &v2)`
User want to install new action for the signal and also User want to collect the current action