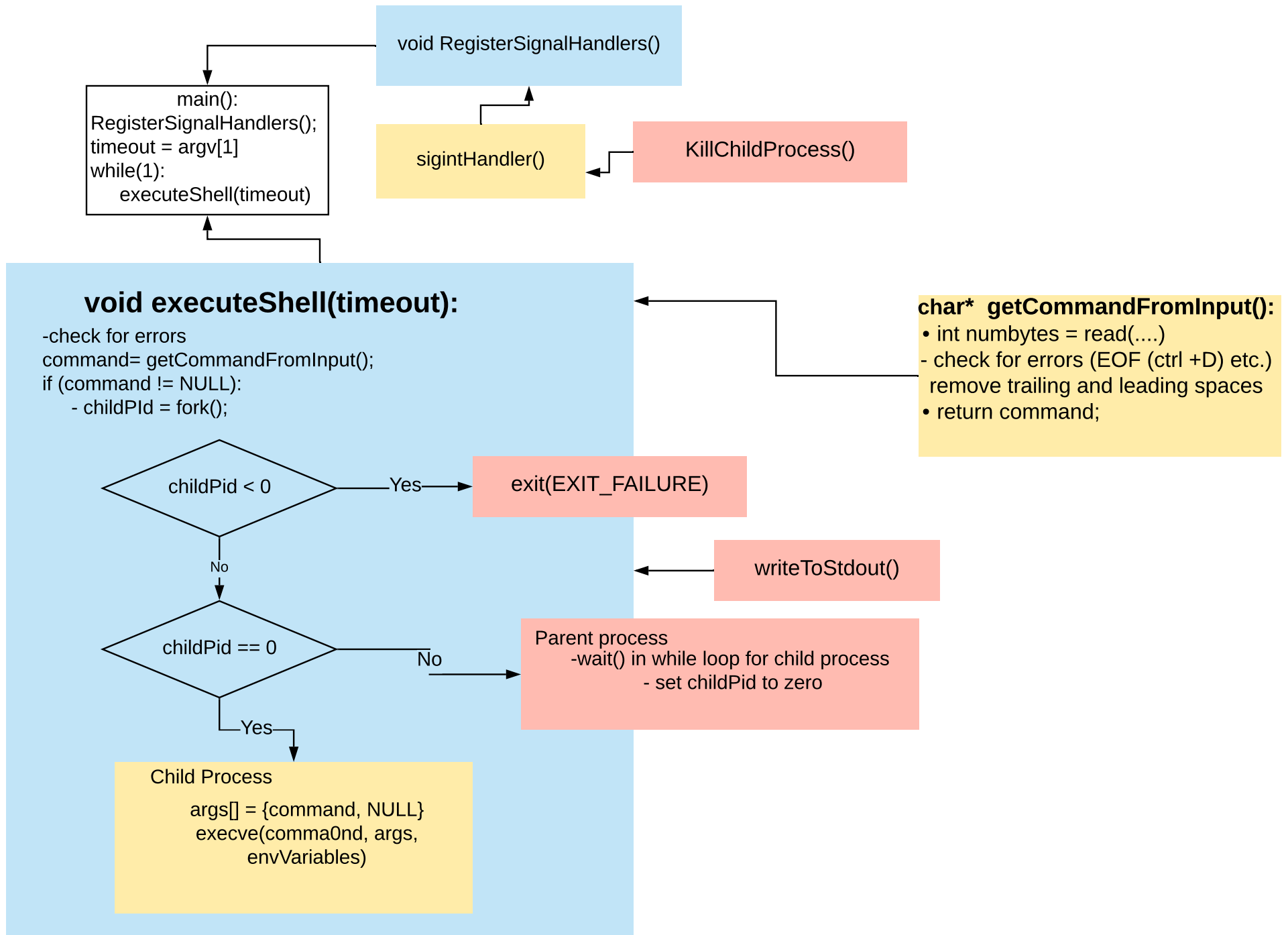


PROJECT 1a



PROJECT 1b

```
main():
RegisterSignalHandlers();
timeout = argv[1]
while(1):
    alarm(0)
    executeShell(timeout)
```

```
void RegisterSignalHandlers();
- signal (SIGINT, sigintHandler);
- signal (SIGALRM, alarmHandler)
```

```
sigintHandler()
```

```
KillChildProcess()
```

```
void alarmHandler(int sig)
```

```
if (sig == SIGALRM)
```

Yes

```
if (childPid != 0)
```

Yes

```
killChildProces();
writeToStdout(text);
```

```
writeToStdout()
```

void executeShell(timeout):

```
-check for errors
command= getCommandFromInput();
-when input is only spaces return so that we can go to new
penn-shredder
-alarm(timeout)
if (command != NULL):
    - childPid = fork();
```

```
childPid < 0
```

Yes

```
exit(EXIT_FAILURE)
```

No

```
childPid == 0
```

No

Parent process

```
-wait() in while loop for child process
- set childPid to zero
```

Yes

Child Process

```
args[] = {command, NULL}
execve(command, args,
envVariables)
```

char* getCommandFromInput():

- int numbytes = read(...)
- check for errors (EOF (ctrl +D) etc.)
- remove trailing and leading spaces
- return command;