

## 2 Process Control

1. 

```
void exit(int status);
void _exit(int status);
void _Exit(int status);

int atexit(void (*function) (void));
```
2. 

```
char *getenv(const char *name);
int putenv(const char *str);
int setenv(const char *name, const char *val, int overwrite);
int unsetenv(const char *name);
```
3. 

```
int setjmp(jmp_buf env);
void longjmp(jmp_buf env, int val);
```
4. 

```
pid_t fork(void);
pid_t vfork(void);

pid_t getpid(void);
pid_t getppid(void);
```
5. 

```
pid_t wait(int *statloc);
pid_t waitpid(pid_t pid, int *statloc, int opt);

opt: WCONTINUED WUNTRACED WNOHANG
macros (termination status):
WIFEXITED(statloc)
    WEXITSTATUS(statloc)
WIFSIGNALED(statloc)
    WTERMSIG(statloc)
    WCOREDUMP(statloc)
WIFSTOPPED(statloc)
    WSTOPSIG(statloc)
WIFCONTINUED(statloc)
```
6. 

```
int execl(const char *path, const char *arg, ...);
int execlp(const char *file, const char *arg, ...);
int execl_e(const char *path, const char *arg, ..., char *const
    envp[]);

int execv(const char *path, char *const argv[]);
int execvp(const char *file, char *const argv[]);
int execvpe(const char *file, char *const argv[], char *const
    envp[]);
```
7. 

```
uid_t getuid(void);
uid_t geteuid(void);
gid_t getgid(void);
gid_t getegid(void);

int setuid(uid_t uid);
```

```
int seteuid(uid_t euid);  
int setgid(gid_t gid);  
int setegid(gid_t egid);
```

8. **int** system(**const char** \*command);

9. pid\_t getpgrp(**void**);  
pid\_t getpgid(pid\_t pid);  
**int** setpgid(pid\_t pid, pid\_t pgid);  
  
pid\_t getsid(pid\_t pid);  
pid\_t setsid(**void**);