

implement os::exec::peek{,any}

From: Bor Grošelj Simić <bor.groseljsimic@telemach.net>

Date: Fri, 25 Feb 2022 01:46:13 +0100

Signed-off-by: Bor Grošelj Simić <bor.groseljsimic@telemach.net>

```
---
os/exec/process+freebsd.ha | 20 ++++++-----
os/exec/process+linux.ha   | 20 ++++++-----
2 files changed, 38 insertions(+), 2 deletions(-)

diff --git a/os/exec/process+freebsd.ha b/os/exec/process+freebsd.ha
index 9601cef8..2365a3c4 100644
--- a/os/exec/process+freebsd.ha
+++ b/os/exec/process+freebsd.ha
@@ -111,7 +111,7 @@ export fn waitall() (uint | error | !(process, exit_status)) = {
 export fn peek(proc: *process) (status | void | error) = {
     let ru: rt::rusage = rt::rusage { ... };
     let st: status = status { ... };
-    match (rt::wait4(*proc, &st.status, 0, &ru)) {
+    match (rt::wait4(*proc, &st.status, rt::WNOHANG, &ru)) {
     case let err: rt::errno =>
         return errors::errno(err);
     case let pid: int =>
@@ -126,6 +126,24 @@ export fn peek(proc: *process) (status | void | error) = {
     return st;
 };

+// Waits for the first child process to complete, then returns its process info
+// and status
+export fn peekany() ((process, status) | void | error) = {
+    let ru: rt::rusage = rt::rusage { ... };
+    let st: status = status { ... };
+    match (rt::wait4(-1, &st.status, rt::WNOHANG, &ru)) {
+    case let err: rt::errno =>
+        return errors::errno(err);
+    case let pid: int =>
+        switch (pid) {
+        case 0 =>
+            return;
+        case =>
+            return (pid, st);
+        };
+    };
+};

+// The exit status code of a process.
export type exited = int;

diff --git a/os/exec/process+linux.ha b/os/exec/process+linux.ha
index ed72fc12..c42bfe4f 100644
--- a/os/exec/process+linux.ha
+++ b/os/exec/process+linux.ha
@@ -97,7 +97,7 @@ export fn waitall() (uint | error | !(process, exit_status)) = {
 export fn peek(proc: *process) (status | void | error) = {
     let ru: rt::rusage = rt::rusage { ... };
     let st: status = status { ... };
-    match (rt::wait4(*proc, &st.status, 0, &ru)) {
+    match (rt::wait4(*proc, &st.status, rt::WNOHANG, &ru)) {
     case let err: rt::errno =>
         return errors::errno(err);
     case let pid: int =>
@@ -112,6 +112,24 @@ export fn peek(proc: *process) (status | void | error) = {
     return st;
 };
```

```

+// Waits for the first child process to complete, then returns its process info
+// and status
+export fn peekany() ((process, status) | void | error) = {
+    let ru: rt::rusage = rt::rusage { ... };
+    let st: status = status { ... };
+    match (rt::wait4(-1, &st.status, rt::WNOHANG, &ru)) {
+    case let err: rt::errno =>
+        return errors::errno(err);
+    case let pid: int =>
+        switch (pid) {
+        case 0 =>
+            return;
+        case =>
+            return (pid, st);
+        };
+    };
+};
+
+ // The exit status code of a process.
+export type exited = int;

```

--

2.35.1



